1986

Hardware design for intelligent vehicles /

Craig Alan Marshall
Lehigh University

Hardware Design for an Intelligent Vehicle

by

Craig Alan Marshall

A Thesis

Presented to the Graduate Committee

of Lehigh University

in Candidacy for the Degree of

Master of Science

in

Electrical Engineering

Lehigh University

1986

CERTIFICATE OF APPROVAL

This thesis is accepted and approved in partial fulfillment of the requirements for the degree of Master of Science.

May 16. 1986
Date

Professor in Charge

Chairman of Department

Acknowlegements

First, and foremost, I would like to thank Professor Nikolai Eberhardt who is responsible for almost all aspects of the vehicle system concept, design, and even the vehicle itself. It has been both a privilege and an incredible learning experience working with him on this project. I am also grateful to Professor Meghanad Wagh for all of the computer-oriented help, especially for the design and programming of the Ground Navigator Board(s).

In addition, I thank SI Handling Systems, Inc. and the State of Pennsylvania for providing the funding to make this project possible.

Finally, I would like to thank my fiance, Elaine, for being patient and putting up with me during the writing of this thesis.

Table of Contents

List of Figures

List of Tables

Abstract

The hardware considerations for an intelligent, autonomous vehicle focus primarily on position determination and navigation. The vehicle system described here is made up of two complementary navigation systems, an optical system and a 'ground' system. The optical navigation system employs an optical goniometer which emits an infrared laser beam that will scan the local environment for passive reflecting beacons. Using the angles at which the beacons are sighted and the associated beacon x,y coordinates, an accurate position 'fix' may calculated by the vehicle's on-board computer. The ground navigation system uses information from a wheel encoder and gyroscope to frequently update vehicle position with its dedicated processor board. Both navigation systems interact with one another to minimize error in position determination.

The central hardware component is the on-board computer, an Intel 86/05 single board computer which coordinates the navigation system functions and provides data to control vehicle movement. All information to and from the on-board computer is passed through peripheral circuit boards which convert data to the appropriate form required by the computer or external hardware.

The vehicle path is prescribed by a base station controller via a digital radio link as a set of x,y coordinates to be traversed. The on-board computer refines the path coordinates and may then proceed along the intended path. Correction for deviations from this path are made as the vehicle travels to permit the vehicle to accurately reach its destination.

Chapter 1
Introduction

Many of today's manufacturers and distributors are turning toward automation as a means to increase efficiency in factories and distribution centers. An automated factory; consisting mostly of computer controlled robotic equipment and very few, if any, human laborers, can increase throughput substantially. In addition, goods of consistent quality may be produced usually at a lower cost than those from a conventional factory. In an environment where material handling is of concern, a cost effective method of moving material about a factory is an automated guided vehicle (AGV) system. An AGV is a driverless vehicle usually designed to pull or carry material from one location to another in a factory. AGVs currently in use typically follow current bearing wires in the factory floor. Not only do the wires provide a guidepath for the vehicles, they also serve as a communication link between the vehicles and a controller. While wire guided AGV systems have proven to be very reliable, they do suffer from a lack of flexibility in that the vehicles are confined to a set of *fixed* paths, and additonally, whenever a path must be altered or added (e.g. to allow for service to new machinery), a certain amount of system down time will be required for installation of new paths. Obviously, a system where the vehicles could be guided by computer instructions to follow any prescribed path would eliminate the constraints on motion imposed by a wire-guided system.

This thesis is intended to serve as a description of hardware design theory and implementation of an optically based navigation system for AGVs. While several non-wire guided navigation systems are currently under development at

several university and industrial laboratories, many of the designs are very complex, requiring sophisticated vision systems and large on-board computational capabilities which do not make these systems commercially feasible at this time. However, when they do become available, these systems will represent the second generation of AGVs with enough flexibility to perform difficult vehicle maneuvers such as navigation around obstacles blocking an intended path.

The navigation system described here bridges the generation gap by being comparable in function to a wire guided system, yet its performance is enhanced by the fact that it is not wire guided and therefore may follow any path prescribed by a base station controller. In addition, this system could be developed to be commercially feasible in a relatively short period of time.

The laboratory demonstration vehicle; Cyclopion, pictured in Fig. 1-1, is used as a test vehicle for the navigation system hardware and software. The system is based around an optical goniometer which sweeps an infrared laser beam around a factory to detect passive beacons situated in the immediate area. By determining the angles at which three of the beacons are sighted, the on-board computer can calculate the position of the vehicle very accurately (beacons sighted at a distance of 10 meters correspond to a maximum of 6 mm error in position). The vehicle also uses a second navigation system consisting of a gyroscope and wheel encoder to provide frequent updates of position and to allow the vehicle to travel a certain distance without relying upon the goniometer information. Signals from the two navigation systems are processed by several peripheral boards containing digital circuitry which converts the signals to a form that can be used by the on-board computer. An Intel 86/05 single board computer is used as an on-board computer to coordinate all vehicle

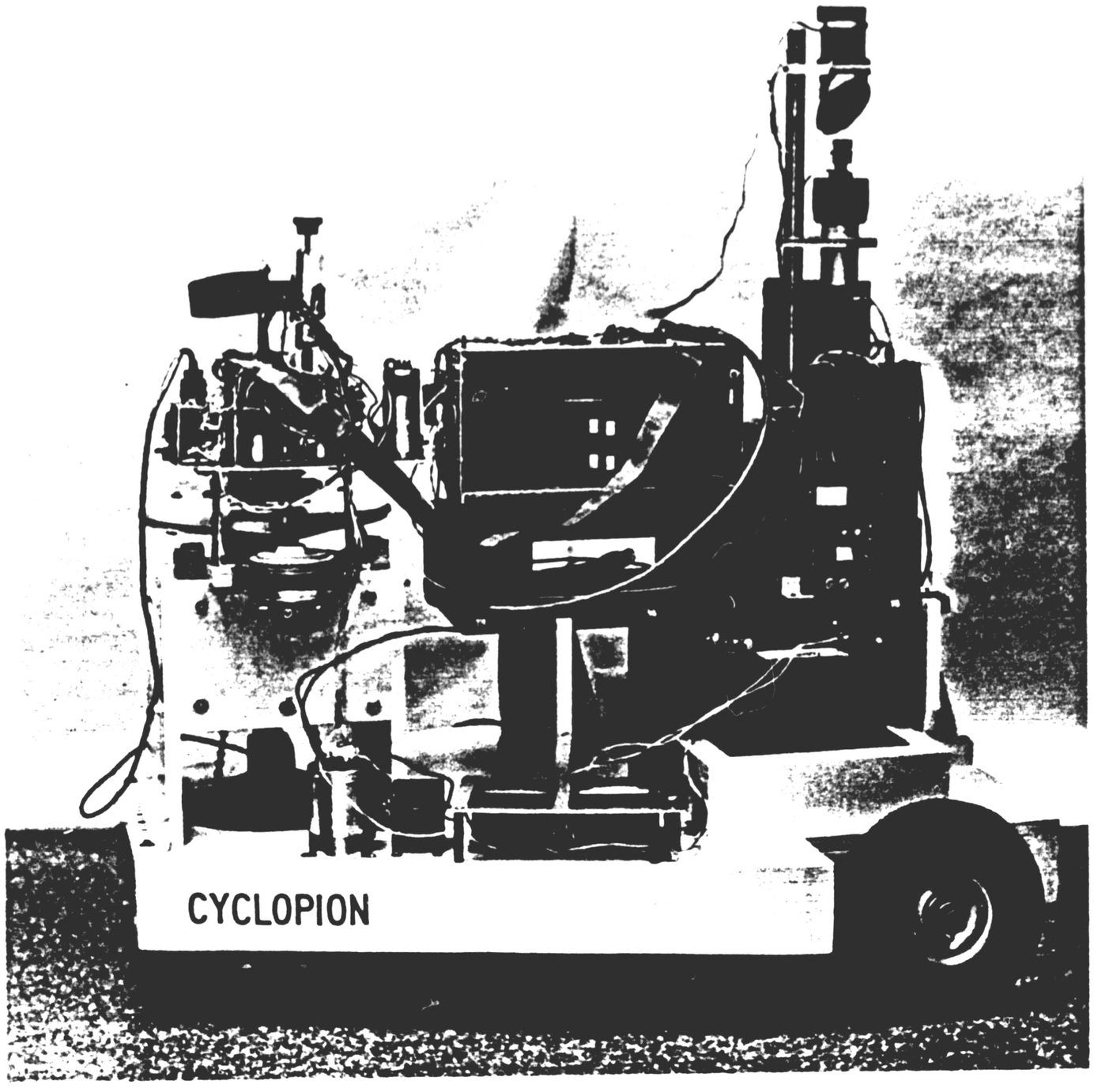


Figure 1-1: Cyclopion, a laboratory demonstration automated vehicle.

functions and control vehicle movement. The 86/05 board is based around the 8086 microprocessor and also contains an additional Intel 8087 numeric coprocessor to handle computations. The vehicle contains a sufficient amount of intelligence so that communications via a digital radio link with the base station computer are kept to a minimum.

Figure 1-2 shows a block diagram of the prototype hardware. A brief explanation of the specially designed peripheral boards is given below:

- **The Signal Conditioning Board-** "Multiplies" the goniometer shaft encoder output for increased angle resolution. It also conditions the

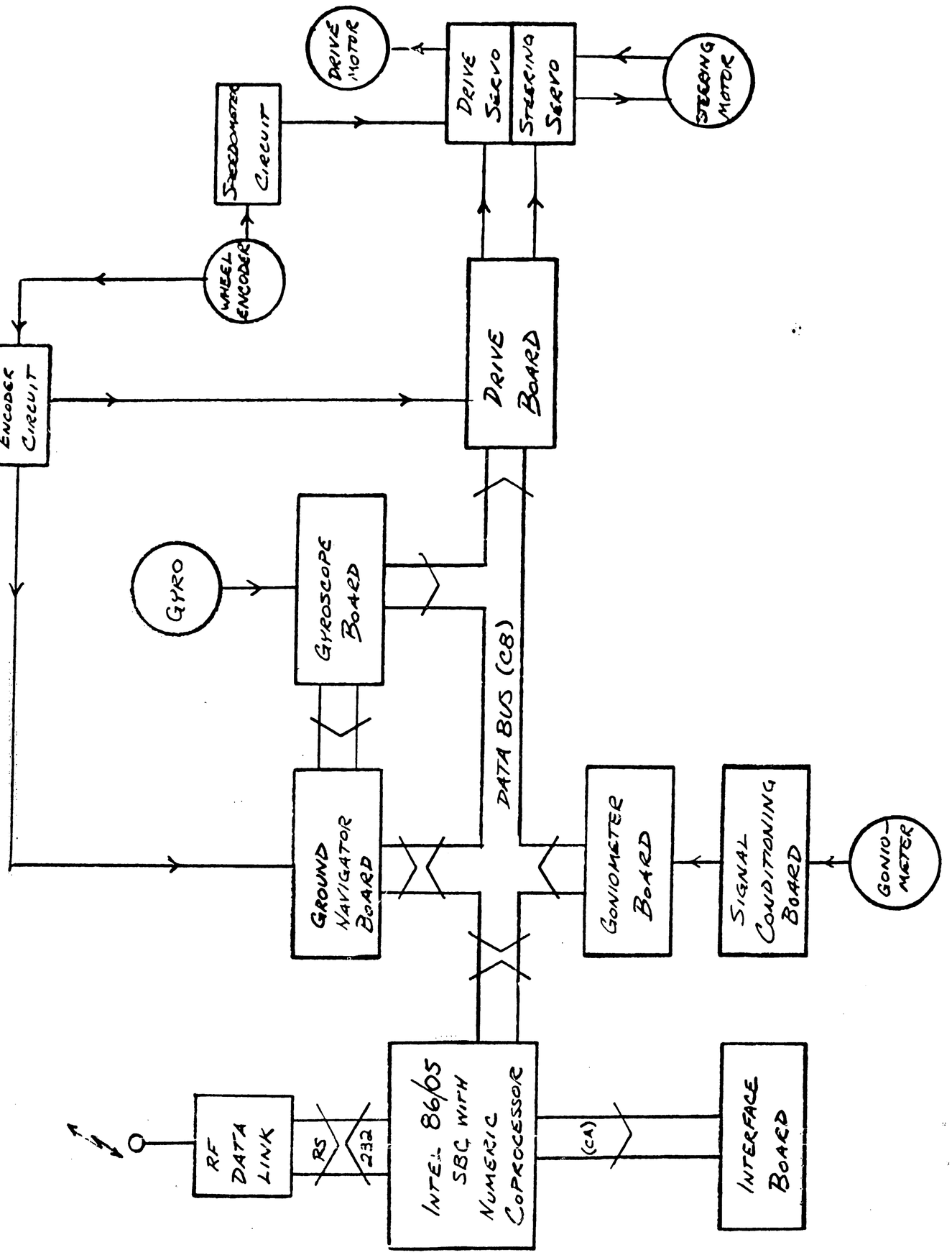


Figure 1-2: Block Diagram of System Hardware.

output of the goniometer photodetector (the beacon 'burst') and uses this signal to create a pulse exactly at the middle of the burst thus establishing the beacon angle.

- **The Goniometer Board**- Counts all beacon angles and beacon widths, writes the counts into temporary memory and facilitates communication of these data to the on-board computer.

- **The Gyroscope Board**- Uses the gyroscope output to continuously track the heading of the vehicle and provides this data to the on-board computer and Ground Navigator Board with a resolution of 1/4 degree.

- **The Interface Board**- Primarily sends control signals selected by the on-board computer to the peripheral boards so that all vehicle data transfers may be coordinated.

- **The Drive Board**- Uses distance and steering angle data from the computer to set the drive wheel direction via an analog output and alerts the computer to calculate the next path segment. It also sets the vehicle speed via an analog output.

- **The Ground Navigator Board**- Functions as a dedicated processor board for the ground navigation system using an 8085 microprocessor to frequently compute and update the vehicle position using data from the gyroscope, wheel encoder, and corrections from the on-board computer.

The hardware described here has been specifically developed for the Cyclopion navigation system. Other approaches to non-wire guided vehicle designs may be found in [5] and [6].

Chapter 2
General Navigation System Design

In general, there are two problems to be overcome for an AGV navigation system that does not use wire guidance. First, the vehicle must know, with accuracy, its location and bearing. Precise positioning is imperative for an AGV especially for docking procedures in which some form of computer controlled robotic machinery will need to act upon the vehicle's load. Second, it needs to know how to get from one location to another in a 'business-like' manner. In other words, vehicle travel should be smooth and the path taken should be as direct as possible. This system addresses both problems.

2.1 Development Goals

In developing this optically based navigation system, several vehicle specifications had to be considered. These were [1];

1. The vehicle will be of the tricycle form, driven and steered by a single front wheel.

2. The vehicle should be capable of moving either forward or backward.

3. The vehicle should be a driverless vehicle capable of following any arbitrary path designated by a base station computer.

4. Deviation from the prescribed path should not be more than $\pm 1/2$ inch at slower speeds to allow for precision docking.

5. The main navigation system will be optical, however the vehicle should contain a secondary navigation system (which is conceivably less accurate) so that the vehicle may continue to follow its path in the event that optical navigation data is temporarily unavailable.

6. Communication between vehicles and the base station should be minimized to avoid overloading the capabilities of the base station computer and to reduce the possibility of error in radio data transmission in areas with a high level of electromagnetic interference.

2.2 Navigation Systems

As was previously mentioned, the vehicle uses two navigation systems. One is an optical system consisting of a goniometer (an angle measurement device), a set of passive beacons, and the associated hardware and software. The second system, the 'ground' navigation system, consists of a gyroscope and a wheel encoder for determination of bearing and distance traveled, and its associated hardware and software.

While the use of two navigation systems may at first seem redundant, both are necessary and actually work together to provide the vehicle with frequently updated, accurate position data. The optical navigation system can determine the location of the vehicle with accuracy and therefore is used for navigation purposes as often as possible. This system, however, may become temporarily impaired if too many of the beacons are obscured by objects or severe dusting (a light dusting of the beacons has shown only a small decrease in the reflected signal). The main disadvantage, however, of the proposed optical system is that it can only supply a positon 'fix' at half second intervals which is not frequent enough for proper navigation. The ground navigation system, while less accurate than the optical system, is capable of providing almost continuous updates of position data to the on-board computer so that the vehicle may stay on its intended course between optical system location fixes.

To obtain the maximum position accuracy possible, the two navigation systems are able to interact with one another. Determining the amount of movement between beacon sightings is critical for the optical system to compute reliable position fixes. This type of data is provided to the optical system by

the ground system. On the other hand, position data, as calculated by the ground navigation system, becomes error prone over long distances, and therefore optical system position data is supplied to the ground system at every location fix to minimize the error.

2.2.1 The Optical Navigation System

The optical system involves the use of retroreflecting surfaces in the area in which the vehicle will be traveling and an optical goniometer (i.e. an angle measurement device) on board the vehicle. The retroreflecting tape reflects light back on itself regardless of what angle the light is incident on the tape. The tapes are wrapped to form a cylindrical shape so that the apparent width is the same from any viewing angle. These retroreflecting cylinders become 'beacons' which are appropriately placed about the factory and their positions recorded in the vehicle memory so that the vehicle may determine its position.

The goniometer is designed to sweep an infrared laser beam around an area where the beam will strike the beacons. The beacons then reflect the beam back to the goniometer to create an electrical signal that is used to determine the angle at which the beacon was sighted. The angles associated with only of three beacons need be used to calculate the position of the vehicle.

As shown in Fig. 2-1, the goniometer consists of an Amperex Electronics Corporation CQL16 diode laser and collimator pen. This laser is modulated at 100 kHz and is run at low power (1 mW, CW) for safety reasons and to extend the lifetime of the laser diode. Modulating the beam enables it to be distinguished from other sources of infrared light so that errors due to spurious signals may be eliminated. The laser is modulated by the Laser Driver Circuit shown in Fig. A-1(a) using a 100 kHz crystal. The current that drives the

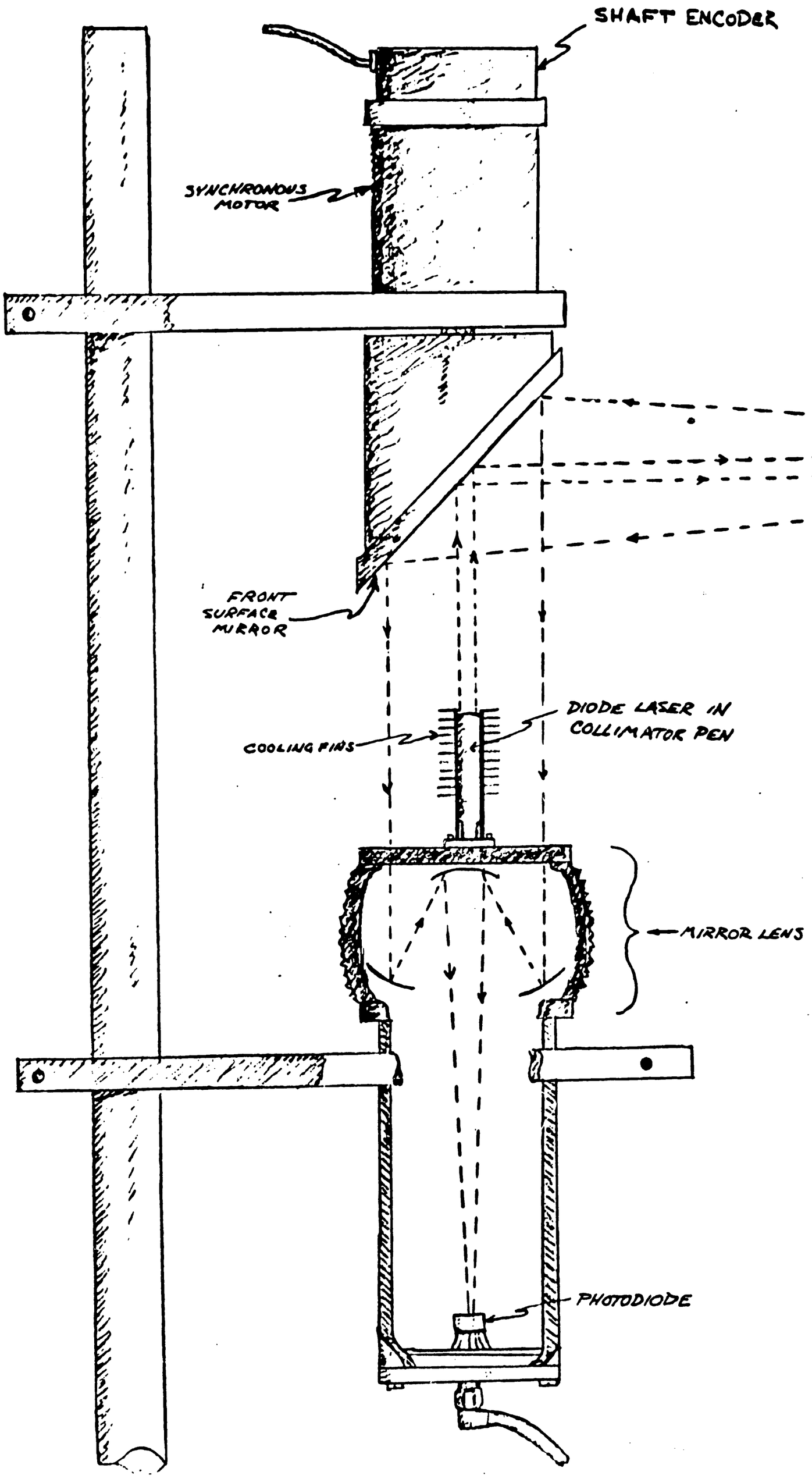


Figure 2-1: The Optical Goniometer

laser is controlled by a feedback loop utilizing the photodiode contained inside the collimator pen. After setting the threshold level by R6, the photodetector section of the circuit uses the laser light to vary the voltage supplied to the 7404 inverter chip. This varies the amplitude of the oscillating voltage to control the current delivered to the laser diode. The laser pen produces a well collimated beam about 5 mm in diameter which is directed upward to strike a front surface mirror. The mirror is situated at 45° to the beam so that it reflects the beam out at 90° to its incidence. This mirror is connected to a synchronous motor which rotates the mirror at 2 rev/sec to sweep the laser beam around in a plane parallel to the floor.

Whenever the laser beam strikes a retroreflecting tape (a beacon), the beam is reflected back on itself. The reflected beam is larger in diameter than the incident beam from the laser pen and therefore much of the light strikes the front surface mirror to be reflected down around the laser pen and into a mirror lens. The mirror lens focuses the reflected light onto an infrared photodiode whose output voltage oscillates due to the modulated laser light. The photodetector circuit, shown in Fig. A-1(b), contains an EG&G HUV 1100BG photodetector whose output is amplified. The resulting output is the beacon 'burst', and indicates that a beacon is being sighted.

Mounted on top of the motor that rotates the mirror is a Hewlett Packard HEDS-6000 Series shaft encoder. The encoder produces 1000 pulses and an index pulse for each revolution of the mirror. The index pulse is used as a zero reference angle for the goniometer and the 1000 cpr output is multiplied on the Signal Conditioning Board by a phase lock loop circuit and used to clock counters to determine the angle through which the motor shaft

has turned. These two signals and the beacon burst are used to determine the angles at which the beacons are sighted.

2.2.2 The Ground Navigation System

A second navigation system has been introduced for two reasons. In many cases, especially in a factory environment, the beacons that the goniometer needs to locate the vehicle may be obstructed. Such situations will rely on the ground navigation system to track vehicle movement. Secondly, the optical navigation system can only determine position every half-second and therefore the ground navigator will be used to locate the vehicle between these (and possibly larger) time intervals.

The components of the ground navigation system are a gyroscope, a wheel encoder, and a dedicated processor, called the Ground Navigator Board, which uses the angle and distance data to frequently update the current position. This system will allow the vehicle to travel relatively large distances without relying on the goniometer. Unfortunately, the location determination error due to gyroscope/encoder navigation increases the longer this navigation method is used. To minimize the error, the position coordinates of the ground navigator are updated with more accurate ones from the optical system each time a position fix is computed from goniometer data. A detailed discussion of the method in which the ground navigator calculates position coordinates may be found in Chapter 8.

2.3 Overall Navigation Routine

The path that a vehicle is to follow is designated by a set of points sent to the vehicle via a digital radio link indicating where any change of direction is required. Between any two points, the vehicle should follow a straight line. This is the form of path sent to a vehicle from the base station computer, however it is too crude for the vehicle to follow accurately. Therefore, the vehicle's on-board computer goes into a software routine which creates a set of secondary points along the proposed path which the vehicle may follow through a series of straight line and circular arc path segments. The vehicle will then proceed along the path using drive data supplied by the on-board computer for each of the path segments.

While in motion, the on-board computer uses the position coordinates provided by the ground navigator (and corrected by the optical navigator) to determine the path being executed and compares it with the intended path. The drive software updates path segments to be followed so that corrections for deviations from the intended path may be made and the vehicle will stay on course. By using this autocorrection scheme, factors such as wheel slippage, tire wear, and direction error due to sluggish steering response need not be considered. If a situation occurs where the error between the current path and the intended path exceeds some nominal value, the vehicle will stop and alert the base station that it is lost and will require reorientation.

The following chapters provide a detailed look at the hardware of the vehicle's on-board computer system. All circuit diagrams may be found in Appendix A. For more information on the software and its design concepts for this system, consult [1], [2].

Chapter 3
The Signal Conditioning Board

The Signal Conditioning Board actually contains two circuits, a phase lock loop circuit and a conditioning circuit. The phase lock loop circuit uses the square wave output from the shaft encoder mounted on the goniometer's synchronous motor as a reference frequency in order to create much higher frequencies that increase the resolution of the angle measurements. This circuit creates two square wave outputs, designated f_c and $f_c/2$, which have frequencies of 200 kHz aand 100 kHz respectively. These outputs are used to create 'half-burst' pulses (which will be described shortly) and the $f_c/2$ output is also used to clock angle and width counters on the Goniometer Board.

The signal conditioning circuit is designed to create 'half-burst' pulses whose trailing edges indicate the center of a beacon burst. This is where the precise position of the beacon is considered to be and therefore it is at these trailing edges that information about the particular beacon is latched into memory. It should be noted, however, that the burst can cause some problems due to its inconsistencies. If a beacon is partially obstructed near its center, two short bursts would occur, implying that two beacons are present. On the other hand, if a burst occurs because of an infrared source other than a beacon (e.g. sunlight or spurious reflections), then erroneous beacon sightings would occur. The signal conditioning circuit can avoid spurious readings such as these.

3.1 The Phase Lock Loop Circuit

The shaft encoder on the goniometer produces a 1000 cycles per revolution square wave. However, to increase the resolution of the encoder, a phase lock loop circuit is used to create a 50,000 cpr square wave. Since the goniometer mirror revolves at about 2 rev/sec, the shaft encoder output is a 2 kHz signal. Referring to Fig. A-2, this signal is brought into U30, a CD4046 phase lock loop chip, whose voltage controlled oscillator has been set to produce a 200 kHz signal which is f_c. The 200 kHz signal will follow the frequency variations of the 2 kHz input if it is divided by 100 and the resulting 2 kHz signal is used as a comparison signal. To do this, the VCO output is used to clock U31a. a CD4520 counter, where the frequency is divided by two at output O_o which provides the $f_c/2$ output of the circuit. The 100kHz output from pin O_o is then divided by 25 by U32 and U33 which are CD4029 counters that have been cascaded and set to count from binary 231 to their maximum count of binary 255 before issuing a terminal count pulse and being reset to 231. The resulting 4 kHz signal from the counters is used to clock one of the two flip flops of U34, a CD4027 dual J-K flip flop, at pin 3. The output of this flip flop toggles at 2 kHz to produce the desired comparison signal for the phase lock loop. The other flip flop of U34 is also clocked by the terminal count pulses from U33, at pin 13, and is used to reset U32 and U33 to binary 231 each time they reach maximum count.

3.2 The Signal Conditioning Circuit

The signal conditioning circuit uses the goniometer beacon burst as an input. The burst is first passed through an analog conditioning circuit which filters out all signals except those oscillating at 100 kHz to avoid passing signals from infrared sources such as lights and the sun. This circuit, shown in Fig. A-1(c), uses a band filter with a Q-factor of about 20 to allow only 100 kHz signals to pass. These signals are amplified before being sent to the Schmitt inverter to 'square up' the signal. Depending on the setting of R31, which controls the threshold level, the output of the Schmitt inverter is either normally high or normally low when no signal is present. The normally high setting is used here. When a burst does occur, the output is a square wave version of the burst.

After passing through the analog conditioning circuit, the burst is used to clock both flip flops of U21. The inverting output of each flip flop is connected to two counters; U23 and U24 for FF U21a, and U25 and U26 for FF U21b. The outputs of the flip-flops are shown in Fig. 3-1. Each pair of counters are cascaded to act as eight bit ripple counters. The flip flops are used as switches to enable and disable counting. When a burst occurs, at time $\mathbf{T_1}$, U21a resets its output, $\overline{O}$, to enable counters U23 and U24. These counters increment until they reach maximum count of 240 at which time they issue a terminal count pulse which sets U21a to a high output state at $\mathbf{T_3}$ to stop counting and reset the counters to zero until the next burst arrives. Flip flop U21b activates counters U25 and U26 in similar manner, enabling counting at the first rising edge of the burst signal. However, at the active high 'clear direct' pin of U21b, the input is the inverted form of the burst which arrives slightly ahead of the

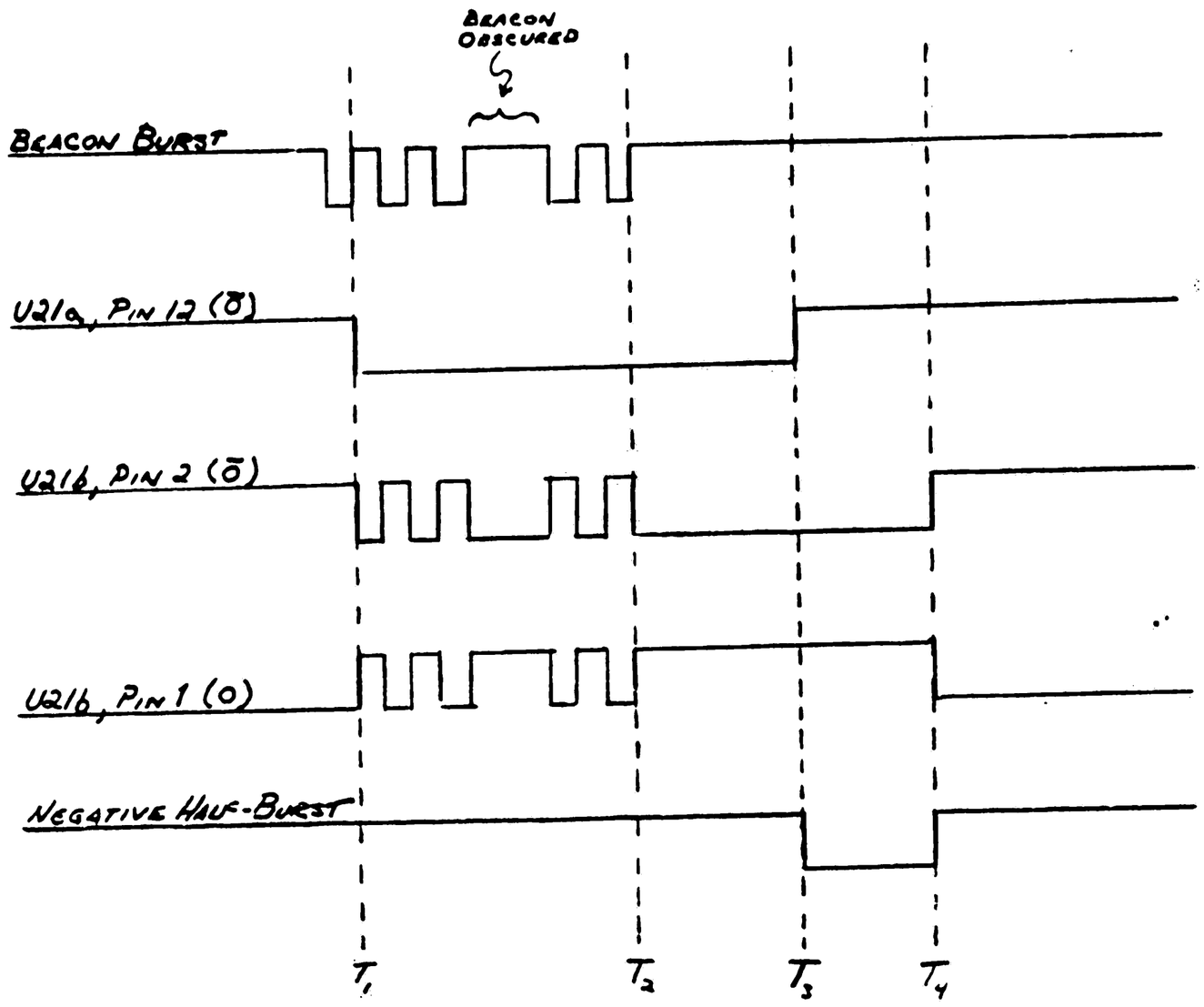


Figure 3-1: Timing of Signals for the Signal Conditioning Circuit

signal at the clock input due to delays introduced in the circuit. This causes the output of U21b to disable the counters (and reset them to zero) at the following falling edge of the burst signal. The counters are enabled and disabled until the end of the burst signal, at time $\mathbf{T_2}$, when they are able to complete their full counting sequence and output the terminal count pulse which stops the process.

It should be noted here that the overall effect of enabling one pair of counters at the beginning of the burst and the other pair at the end of the burst is to create pulses of equal width at the outputs of the flip flops. The difference in the time at which the pulses occur is equal to the time duration of the burst. Therefore the full width of one beacon is determined regardless of whether the burst was continous or not. Thus any beacon which is obscured

between its edges will still be counted as *one* beacon.

Notice that $\overline{O}$ of U21a and O of U21b are NANDed together at U29a. Also notice that during counting, the O outputs of the flip flops U21a and b are high and the $\overline{O}$ outputs are low. Therefore the output of U29a is low whenever U23 and U24 are not counting but U25 and U26 are, which means that the output of U29a should be low for a time equaling the duration of the burst but slightly delayed in time. Actually however, we are more interested in making the output of U29a low for half of the duration of the burst so that the center of the burst may be determined. To do this, the clocking frequency to the counters is switched to 200 kHz, twice the normal frequency, whenever the output of U29a goes low which is shown at time $\mathbf{T_3}$ in Fig. 3-1. This allows the counters to reach maximum count in half of the time and force the output of U29a to remain low for a time equal to half of the burst duration. For this reason, the output of U29a is called the 'negative' half-burst. The inverted form of this pulse is ORed with the index pulse to create output pulses which occur for both index pulse and beacon sightings. These signals will be used by the Ground Navigator Board to initiate the loading of position coordinates into memory (see Chapter 8).

Chapter 4
The Goniometer Board

The Goniometer Board provides data from the optical navigation system to be used for computation of position. Three kinds of data are available from this board; the goniometer angle reading for various beacons, the width of the beacons, and the RAM address of the information to identify a particular beacon. This information is stored in memory during each revolution of the goniometer and is available to be read by the on-board computer at the end of the revolution, if desired. During any one revolution of the goniometer, only beacon information obtained during that revolution is stored.

4.1 The Goniometer Angle

As the laser beam sweeps through one revolution, it will strike several beacons. Each beacon seen will have a particular angle relative to the index pulse from the encoder on the goniometer. The angle given to each beacon will be a binary number between zero and 50,000, where zero corresponds to zero degrees (i.e. at the same point as the index pulse) and 50,000 corresponds to 360°. It should be noted that there is a 'dead' angle in which no beacons may be seen, slightly before and after the index pulse, so that angle readings around 0 and 50,000 will not actually occur. The purpose of the dead angle will be explained later.

Referring to Fig. A-3, the two CD4520 counters, U35 and U36, are wired to perform as a 16 bit ripple counter. These counters form the goniometer angle counter. U35 and U36 are clocked by $f_c/2$, a 100 kHz signal, so that they count from zero to approximately 50,000 on one rotation of the goniometer

before being reset to zero by the index pulse. Therefore these counters will keep track of the angle through which the laser beam is sweeping. The outputs of the angle counters are connected to the inputs of U37, U38, and U39, which are MC14503 tri-state buffers. The outputs of these buffers are held in a high impedance state until the half-burst signals arrive from the Signal Conditioning Board. The trailing edge of the 'positve half-burst or index' signal causes U144a, a monostable multivibrator, to produce a 0.2 μsec pulse which enables the buffers' outputs so that the angle of the current beacon may be written into the RAMs. The details of writing the data into the RAMs will be explained shortly.

4.2 Beacon Width

In addition to storing the angle of each beacon sighted, we also want to store the width of the beacon. Physically, all of the retroreflecting tapes are the same in width; however, the closer the tape is to the goniometer, the larger it appears since the burst will be longer. The burst duration, then, is actually a measure of the apparent width of a beacon. Motivation for storing width data comes from the fact that while only three beacons are needed to calculate the vehicle location, the accuracy with which the position may be determined increases when the beacons are close. Therefore we use the width measurement as a means of determining which beacons may be rejected as being too far away for accurate calculations.

Rejecting beacons by virtue of their width also avoids 'bad' beacon sightings. As was prevoiusly mentioned in Chapter 3, the beacon burst could give false information if a beacon was obscured near its center or if other infrared sources produced spurious signals. These situations were compensated

for in the design of the Signal Conditioning Board. There are however, other situations in which the beacon burst could be misleading, yet by rejecting beacons of small width, the error due to them may be reduced. For example, if the laser beam were to strike a reflective surface, a short beacon burst would occur when the beam was normal to the surface. Similarly, if a beacon is partially obscured at its edge, a shorter than normal burst would occur. (In this case, a beacon that has its edge only slightly covered may be used in computations with only a slight error. The difference between actual beacon center and apparent center is equal to half of the amount of beacon covered.) For both of these cases, taking into account the width of the burst will reduce the possibilities of using such faulty readings.

Since only the relative widths of the beacons are of concern, it is convenient because of the board design to measure the duration of the half-burst instead of the burst itself. The two CD4029 counters, U49 and U50, are connected to produce an eight bit representation of the beacon width. These counters are clocked by $f_c/2$ and counting is enabled only during the half-burst so that the counters count the time duration of the half-burst. The output of the counters is fed into two MC14503 tri-state buffers, U51 and U52. Near the end of the counting cycle, the buffers become transparent so that the data is sent directly to the HM6516 'width' RAM, U46. At the end of the half-burst, counting is disabled, the buffers are set to a high impedance output state, and the most recent number entered into the RAM is latched to become the 'width' of a particular beacon.

4.3 Reading and Writing to the RAMs

To write data to or read data from the Harris HM6516 RAMs on this board, three signals are needed; $\overline{W}$, $\overline{G}$, and $\overline{E}$. $\overline{W}$ and $\overline{G}$ are the write and read enables respectively, and $\overline{E}$ is the chip enable. $\overline{E}$ must go low to latch in the memory address for any read or write operation. The $\overline{W}$ input on each RAM will go low for 0.2 μsec for each beacon burst and the index pulse to allow the proper data to be stored. Lowering $\overline{G}$ starts a reading sequence where all data stored in the previous goniometer revolution may be read from the RAMs. During this read sequence, $\overline{G}$ may remain in the low state.

4.3.1 Writing Procedure

Before looking at the circuit in more detail, it will be helpful to consider the procedure used in writing to this board. As the goniometer sweeps through one revolution, several beacons should be sighted. At each beacon sighting, an address counter will be incremented by one and the goniometer angle and beacon width information will be latched into the RAMs at that address. Since the address counter begins at zero, the first beacon seen would increment the counter to one and the beacon width and angle will be written into the RAMs. Similarly, the second beacon seen will increment the counter to two and its angle and width data will be written. At the end of the revolution, the index pulse will be taken as a beacon sighting, and the address counter will be incremented once more. (The address for information about the index pulse is used on the Ground Navigator Board. Information stored in memory on the Goniometer Board is irrelevant and will not be used.) In all, a maximum of fourteen beacons and one index pulse may be seen in one revolution of the goniometer.

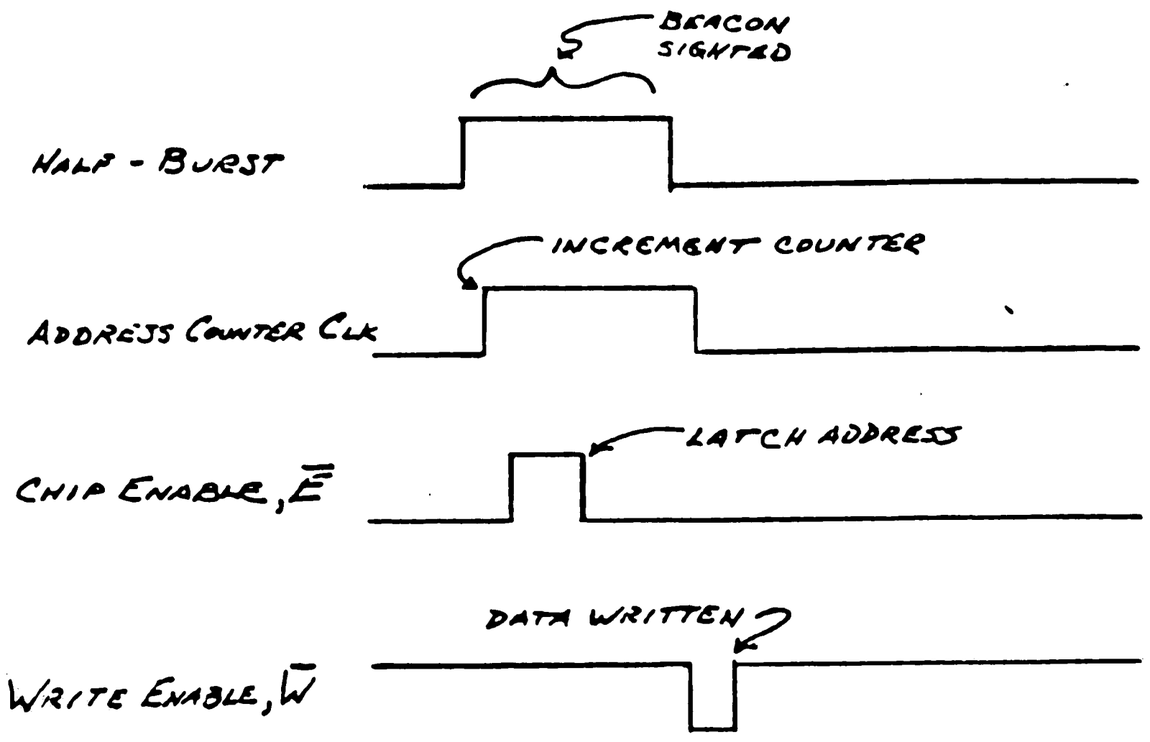


Figure 4-1: Signal Timing Sequence for the Write Procedure

Figure 4-1 shows the timing sequence of the controlling signals involved in the writing procedure. As implied in this figure, the positive and negative half-burst signals from the Signal Conditioning Board initiate the writing process. When a beacon is sighted, the rising edge of the positive half-burst clocks U42. the memory address counter to increment the present address by one. At the same time, the negative half-burst triggers monostable multivibrator U44b to produce a short 0.1 μsec pulse. This pulse is sent to the three RAMs (U40, U41, and U46) at the $\overline{E}$ input so that on the falling edge of the pulse, the current address will be latched.

The positive half-burst input is also connected to monostable multivibrator U114a which will produce the 0.2 μsec pulse used to lower the $\overline{W}$ input on all RAMs and also causes data buffers U37, U38, U39, U51, and U52 to become transparent. This allows beacon angle and width data to be written into the RAMs. The goniometer angle counters run continously and therefore beacon angle information is always avaikable. The beacon width counters, U49 and U50

however, are designed to count only for the duration of a half-burst. This period of time is extended slightly to accommodate the fact that writing to the RAMs is performed just after the half-burst. To accomplish this, the negative half-burst is ANDed with the write enable pulse by U115d and U115c. An RC combination is provided to delay the half-burst pulse at U115d so that no glitch occurs. The output of U155c then, is a low pulse that is slightly longer than the actual half-burst signal. It will be this pulse that is actually counted as the beacon width since when the pulse occurs, the beacon width counters, U49 and U50, will begin counting. At the end of the pulse the count will be latched into memory and the counters will be reset to zero.

4.3.2 Read Mode

The index pulse indicates that the goniometer has swept through one revolution and initiates the read mode for the board, as shown in Fig. 4-2.

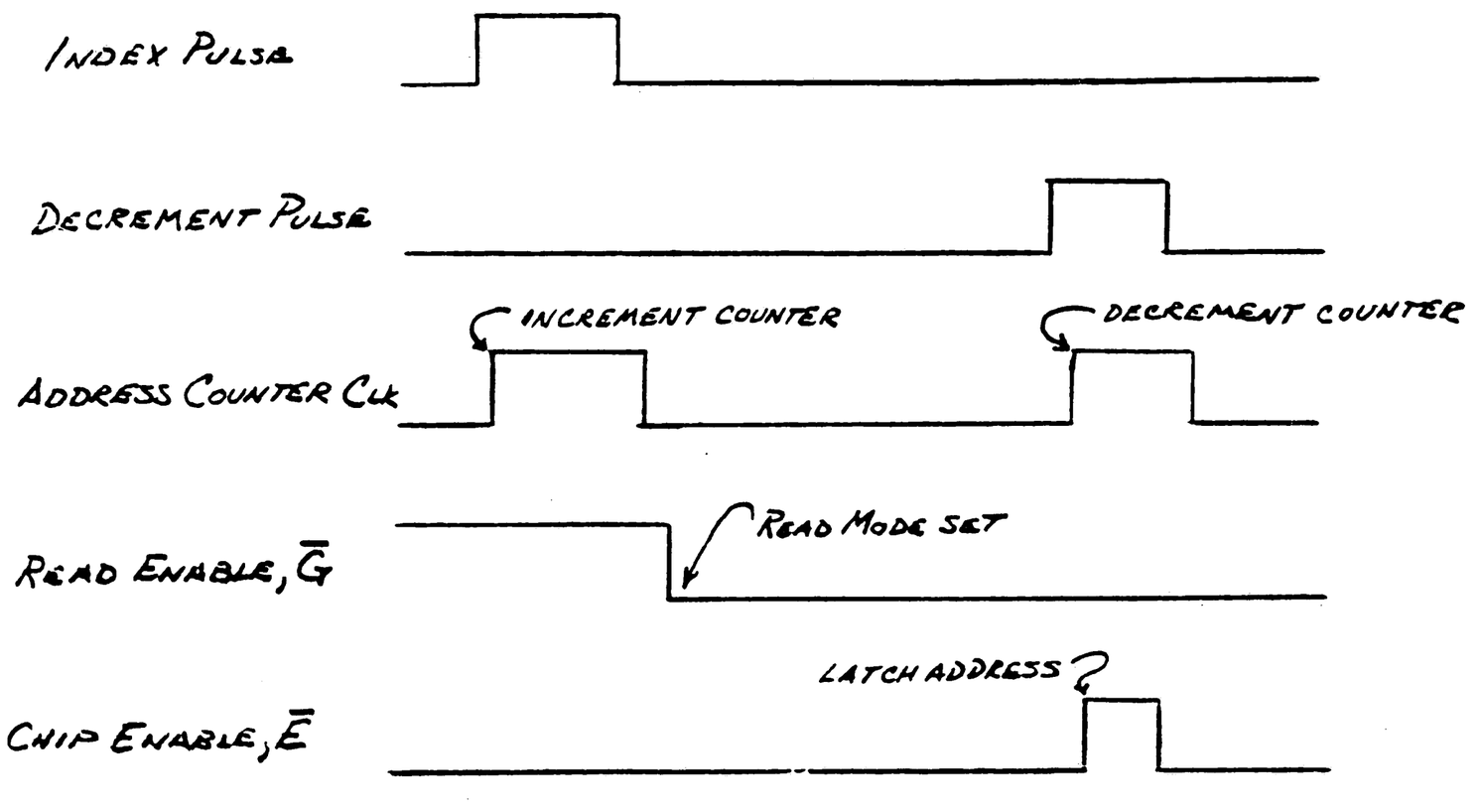


Figure 4-2: Signal Timing Sequence for the Read Mode

When this pulse occurs, several things happen. First, the index pulse triggers two monostable multivibrators. One is U44a which creates a 1 μs delay before clocking flip flop U45a to enable the RAMs to be read. This delay is used to allow the Ground Navigator Board time to compute and store position data at the index address. The index pulse also triggers the pair of monostable multivibrators of U43. These two monos act as a 20 msec timer before automatically resetting the address counter to zero and disabling the read mode. This timing circuit allows sufficient time for the on-board computer to read all of the information in the RAMs before resetting the board.

When the output of flip flop U45a goes low, information may be read from the RAMs. The present address will be the index address and the on-board computer will read information from the Ground Navigator Board RAM but no information from memory on this board since it is irrelevant. The computer will then issue a decrement pulse which reduces the current address by one and also causes monostable multivibrator U44b to produce a 0.1 μsec pulse output for the $\overline{E}$ input on the RAMs so that the new address will be latched.

At this point, the computer will sequentially enable buffers U53, U54, U55, and U56 to obtain angle, address, and width information about the beacon seen *last* in the previous revolution. Data is also available at this time on the Ground Navigator Board which will be read by the computer before another decrement pulse is issued. After the next decement pulse, the computer will read the information about the second to last beacon sighted. This process continues until all beacon information has been read and the address counter has been decremented to zero. The zero output of the address will cause the output of the U47d/U48a NOR logic to go high which will in turn cause the $\overline{O}_1$

output of flip flop U45a to return high. Now the board is out of its read mode.

If the computer does not read information from this board during the read mode, the address counter is automatically set back to zero by U43 and the same process as described above will get the board out of read mode.

4.4 The Dead Angle

The interval in which the Goniometer Board is in its read mode is very important. This is the only time during a goniometer revolution that the on-board computer may read information from the RAMs. During this time period any beacons sighted will not be recorded, hence it is called the 'dead angle' with reference to the goniometer.

The size of the dead angle is dependent upon the amount of time needed for the computer to perform all read operations for the maximum number of beacons plus the delay time associated with beacon burst to half-burst. Whenever a beacon is sighted, a half-burst signal is generated whose trailing edge indicates the center of the beacon. However, the half-burst signal is slightly delayed in time from when the beacon was actually seen. Therefore the time delay associated with the beginning of the beacon burst to the end of the half-burst must be accounted for in terms of a dead angle before the index pulse should occur. Preliminary estimates show that a dead angle of approximately ten degrees will provide sufficient time, about 15 msec, for reading information for all beacons from the RAMs.

Chapter 5
The Gyroscope Board

The Gyroscope Board is designed to accept the output signals from a King Radio Corporation 102A aircraft gyroscope and convert them to an angle. A reference angle, supplied by the on-board computer when needed, may be input to the board via the data bus. The Gyroscope Board produces a twelve bit representation of the vehicle bearing in degrees which consists of a right/left turn bit (msb) and eleven bits representing the current angle in a range of 0° to 360° with quarter-degree resolution.

5.1 Creating the Binary Angle

The input to this circuit from the gyroscope comes in the form of two square waves, one phase shifted by 90 degrees from the other, with amplitudes of 30 volts peak-to-peak. The square wave outputs oscillate between +15V and -15V. When these signals are input to the board, as shown on Fig. A-4, a pair of diodes are used to block out the negative voltages and an 11MΩ series resistance will prevent the +5V operated CMOS gate from overload.

The logic gates of U3 simply act as a latch. When the enable line is high, this section of the circuit is transparent to signals as they pass to the Schmitt NAND gates U4a and U4b. However, when the enable line is low, the outputs of NOR gates U3b and U3d are frozen. This latched state is employed when the buffers are being read by the on-board computer to avoid a change of logic state during the read operation.

Gyroscope signals are used to operate a set of counters that would count the number of degrees turned through; however there is a problem. In general,

shaft encoders suffer from a multiple count problem when a stationary shaft is oscillating slightly about a logic edge. Therefore it is necessary to apply the inputs to the angle counters in such a way as to avoid the errors that could occur as a result of the multiple counting. To do this, the outputs from NOR gates U3b and U3d, labelled E and D in Fig. 5-1, are inverted and then fed into exclusive ORs U5a and U5b. Notice that the inputs to XOR U5a are the same signal except that the signal at pin 5 is inverted and slightly delayed due to the RC combination. This causes the output of U5a to be a series of short negative pulses occuring at each transition of the input signal. The trailing edges of these pulses will be used to clock the three CD4029 counters (U6, U7,and U8) that make up the angle counter. The counters will be set to count up or down, depending on the rotational direction of the

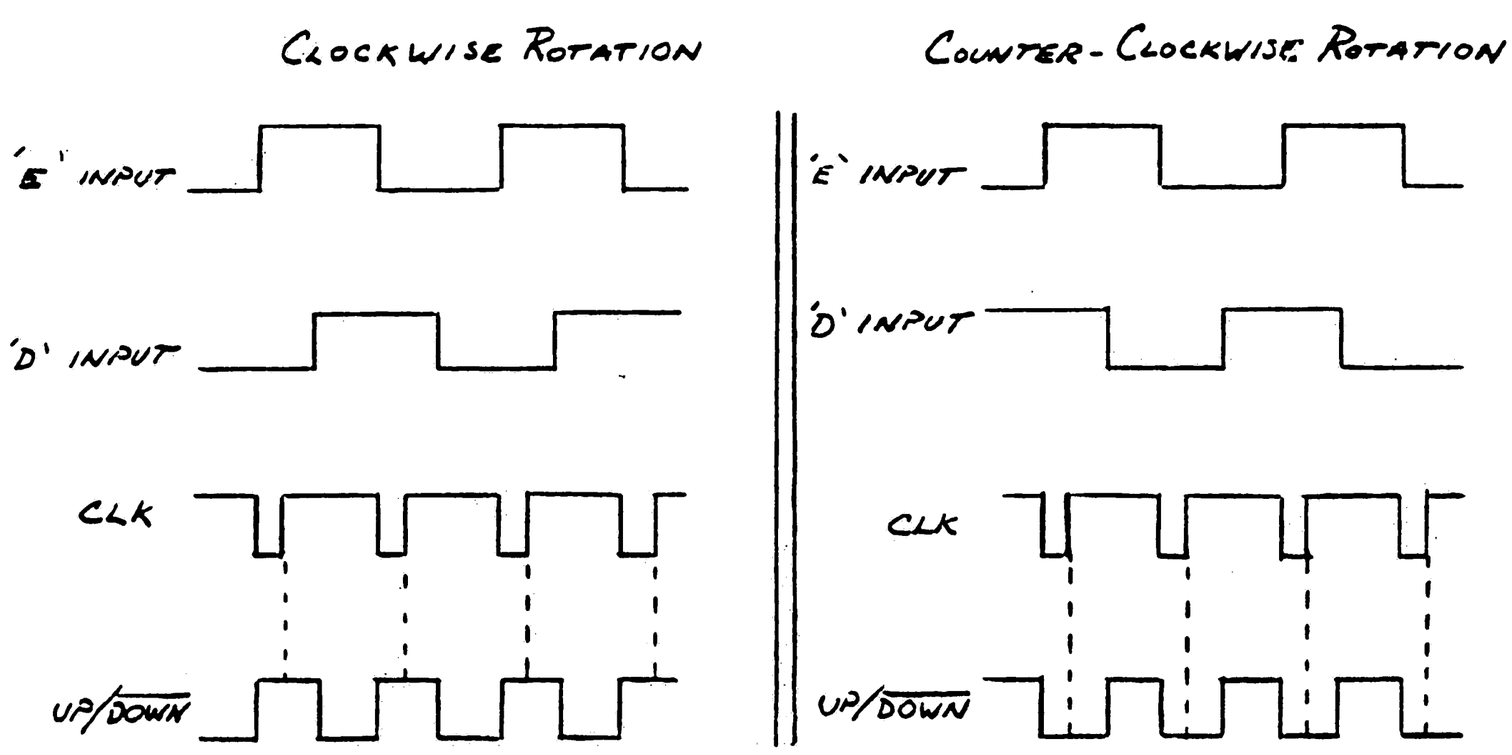


Figure 5-1: Timing Diagram for Clocking Angle Counters, adapted from [4]

gyroscope, by the output of the other XOR gate, U5b. This gate contains, as inputs, the inverted signals from the NOR gates U3b and U3d and its output alternately sets the angle counters to count up and down. Whether the

counters actually do count up or down depends upon their up/down setting at the moment they are clocked by the trailing edge of the short negative pulses from the output of U5a. Thus it may be apparent that this solves the multiple counting problem since an oscillating signal from the gyroscope due to shaft vibrations will cause the counters to alternately count up one and then down one to maintain an accurate reading of the angle [4].

To send the data to the on-board microprocessors, two pairs of 74LS244 tri-state buffers are used; U9/U10 and U120/U121. Each pair of buffers contains the same data consisting of a low byte and four bits of the high byte. U9 and U10 are enabled, one at a time, by the on-board 86/05 computer board onto its eight bit bus when an angle reading is desired. The other pair of buffers is enabled simultaneously onto a 16-bit bus by the 8085 microprocessor on the Ground Navigator Board. It should be noted that the data taken from the buffers is not just the gyroscope angle reading, but actually three pieces of data, set up in the following manner:

- **Bit 11 (MSB)**- '1' indicates right turn being performed. '0' indicates left turn being performed.
- **Bit 10 to Bit 2**- Integer value of angle.
- **Bit 1 and Bit 0**- Half and quarter degree value of angle.

For convenience, it is desired to have the output of the angle counters to be a binary number between 0.0 and 360.0 to indicate the angle of the vehicle. To accomplish this, a change of the angle reading to zero is desired whenever 360 degrees is reached and the angle is increasing. Conversely, if the angle reading decreases to zero, it should be changed to 360.0. The output of U17a, a CD4013 flip flop, is high if the angle is increasing (i.e. a right turn is being

performed), and low if the angle is decreasing (i.e. for a left turn). This is the most significant bit of the data bus and, in combination with the output of the angle counters, is input to both a group of OR gates and comparators. The OR gates U11a and b, and U12a and b, are used as a comparator to indicate that the angle zero has been reached and the angle is still decreasing. If this occurs, the following sequence of events will occur to change the angle reading to 360.0:

Suppose all of the bits on the bus are zero. This will force pin 13 of U12b to go low. The low output will enable the 'A' outputs from U18, an MC14503 tri-state buffer, and will also cause the output of inverter U118f to go low. The inverter output is connected to OR gate U117a whose output will also go low to enable buffers U20 and U119B provided that its other input at pin 2 is low. Therefore U117a prevents enabling of the buffers if the on-board computer is currently sending updated angle information to the angle counters. Also connected to the output of U118f are NANDs U116b, c, and d. Upon a low signal from U118f, all of the NANDs will go high to enable the angle counters to be loaded with data from buffers U20 and U119B. The buffers will output either all zeros or binary 360.0 depending on the output enabled on buffer U18. In this case the 'A' outputs of U18 are enabled and line O_1 will output a one. This will create a binary 360.0 at the inputs to U20 and U119B which will be loaded into the angle counters when these buffers are enabled. Now that the angle counters are showing 360.0, this forces the output of U12b to return high to disable U18A, U119B and U20, and enable the angle counters to count from 360.0. A similar sequence of events occurs if the data bus contains the binary number 360.0 and the MSB is one, indicating an increasing

angle. The three cascaded CD4585s; U13, U14, and U15, are comparators set to output a high signal if this situation occurs. The output is inverted and enables the 'B' outputs of U18 where a zero is sent out on O_6. The same process as described above will allow the angle counters to be loaded with all zeros. An all zero output from the counters causes the comparator output to return low; buffers U18B, U119B, and U20 to return to a high impedance output state, and enables the counters to resume counting from zero.

Whenever a buffer is enabled, the gyroscope input is frozen for 60 μsec, via the 'freeze' input, by a monostable multivibrator on the Interface Board. This avoids any changes in the angle reading while the microprocessors latch the data. The actual time that the gyro input is frozen is sufficiently short to allow the data to be read and yet not cause error due to missing pulses from the gyroscope's output. During that time, not more than one change of state can happen, and upon removal of the disable pulse, such a change would be immediately counted.

5.2 Setting the Gyroscope Angle

To provide a zero reference angle or to update the current gyroscope angle to a more accurate one, the on-board computer selects the high and low byte 'Set gyro' control signals from the Interface Board. The sequence begins with the computer putting the low byte of the new angle on the data bus and then selecting the 'Set Gyro, LB' control signal. The control signal will enable the eight bits to pass to the output of buffer U19. The output lines of the buffer are connected to the parallel data inputs of angle counters U6 and U7 where the data is now present and will be loaded as the control signal causes NANDs U116b and c to go high.

Having loaded the low angle byte, the 'Set gyro, LB' control signal is removed. The high byte of the new angle (of which only the two least significant bits are used) is now placed on the data bus and the 'Set gyro, HB' control signal is selected to go low. This will enable the 'A' buffer outputs of U119 to allow the data to pass to the parallel inputs of angle counter U8. The control signal will also cause the output of NAND U116d to go high so that the data may be loaded into the counter. At this time, the control signal is removed and the new gyroscope angle is set.

Since there is the possibility that there could be a zero or 360.0 angle replacement (as described in the previous section) at any time, it is necessary to prevent this from occurring while the new angle bytes are being loaded. To do this, the two 'Set gyro' control lines are connected as inputs to NAND U116a. If either control signal is selected (i.e., goes low), the output of the NAND goes high so that any enabling of buffers U119B and U20 is prevented by OR gate U117a. In this manner, bus contention of the parallel data input lines is prevented.

Chapter 6
The Interface Board

The Interface Board is used as a link between the on-board computer and the peripheral boards. This board primarily sends out control signals to the peripheral data buffers to allow data to be read from or sent to the boards via the data bus. Some of the other functions available on the board are; the creation of an interrupt pulse to begin the goniometer information reading cycle, creation of the 'gyro data freeze' signal, ability to set the vehicle direction, and the creation of the memory address counter decrement pulse. The functions of all outputs for this board are shown in Fig. A-5.

All functions except the goniometer read cycle interrupt are available to the computer through its Port CA. Each byte sent to this port will select one of the board control functions through the use of two demultiplexing chips, U76 and U77. Since these chips are always enabled, a dummy address of 00H is used when none of the output functions is desired. The 00H address is chosen to avoid probelms which may occur since Port CA will output 00H for a short time whenever the 8255A PPI device on the computer board is reinitialized. This will usually occur when the device is being reprogrammed to switch the I/O status of Port C8 (the data bus). Table 6-1 shows the hex address and names of the functions available on the Interface Board.

Hex Address	Function
00	Dummy
01	Address Counter Buffer Enable
02	Goniometer Angle Buffer Enable, HB
03	Goniometer Angle Buffer Enable, LB
04	Beacon Width Buffer Enable
05	Set Gyroscope Angle, LB
06	Set Gyroscope Angle, HB
07	Δy Strobe Signal, for HB
08	Δy Strobe Signal, for LB
09	Δx Strobe Signal, for HB
0A	Δx Strobe Signal, for LB
0B	X-Coordinate Buffer Enable, LB
0C	X-Coordinate Buffer Enable, HB
0D	Y-Coordinate Buffer Enable, LB
0E	Y-Coordinate Buffer Enable, HB
0F	Decrement Pulse
10	Gyro Buffer Enable, HB
20	Load Primary Distance Latch
30	Load Secondary Distance Latch
40	Load Primary Steering Latch
50	Load Secondary Steering Latch
60	Load Speed Latch
70	Gyro Buffer Enable, LB
80	Set Forward Drive
90	Set Reverse Drive
A0	Coordinate Request

Table 6-1: Interface Board Port Address and Functions

6.1 Setting the Read Mode

Each revolution of the goniometer will write information into the RAMs of the Signal Conditioning Board and the Ground Navigator Board. At the end of one revolution, the index pulse will set the RAMs to be read by the computer. The computer may not, however, be able to read the contents of the RAMs at this time. When the computer is able to read the data, it will set one bit of its output port C, CC_1, low. Then it waits for an interrupt created by the index pulse to begin reading the data. In hardware, this is done using OR gate U79a. The output of this gate will go low to interrupt the computer when CC_1 at pin 8 is low and the index pulse occurs.

6.2 The Read Operation

When the interrupt signal occurs at the INT3 pin on the computer, a read sequence is begun to take in goniometer data from the Goniometer Board, position data from the Ground Navigator Board, and bearing data from the Gyroscope Board. The read sequence will progress as data buffers are selectively enabled by outputs from the computer's CA port to the two DMUX chips. When data from the gyroscope buffers is requested, the monostable multivibrator U80a is triggered to send a 60 μs pulse to the Goniometer Board to freeze the gyroscope inputs while the buffers are being read. This will occur when either the on-board computer or the Ground Navigator Board enables the gyroscope buffers.

6.3 Forward/Reverse Selection

The Interface Board also contains the forward/reverse circuitry. The on-board computer may select either direction of travel via U77. Vehicle direction will be represented at the output of flip flop U78a with a low state indicating 'forward' and a high state indicating 'reverse'. NOR gates U111a and U111b are wired to provide a latched output for direction control that is sent to a 2N3904 transistor which will switch a relay between its open and closed positions. The relay arms are connected to the drive servo to control drive wheel direction.

Chapter 7
The Drive Board

The Drive Board allows the on-board computer to control the driving and steering system of the vehicle. It accepts information from the computer via the data bus specifying vehicle velocity, steering angle, and length of path segment to travel. The wheel encoder output from the front wheel is input to this board to determine when the present path segment distance is almost completely traversed so that the computer can provide additional drive instructions.

7.1 The Drive Routine

As the vehicle travels along its intended path, it must correct for driving error which forces it off this path. Therefore the on-board computer will navigate the vehicle through segments of the overall path by checking for position errors through a path segment and correcting for these deviations with the drive instructions for the following segment. Performing this routine will allow the vehicle to compensate for slippage, tire wear, and steering response time.

The software for computation of path segments allows the computer to specify two proceeding segments based on the postion of the vehicle near the end of the current segment and the intended path. Only the first of the two specified path segments will normally be used so that driving error will be minimized. However, both segments will be used if the first of the segments is so short that there is insufficient time for the computer to generate new path segments. This situation may occur when the vehicle is to position itself to

some prescribed bearing and must perform a short double arc maneuver to do so (such as in docking procedures). In short, by computing and storing two future path segments near the completion of a current segment, drive instructions will always be available for the vehicle.

7.2 Hardware Implementation for the Drive Theory

A small Micro-Mo Electronics 125 pulses per revolution shaft encoder is connected to the driving wheel motor. The gear ratio from motor to drive wheel will allow one pulse for each 0.01 inches of travel. As seen on Fig. A-6, the output of the encoder is squared up by U74a, a Schmitt NAND, and will be used to signal the Ground Navigator to compute the present vehicle location every 0.01 inches.

The encoder output is also connected to a pair of CD4029 counters, U57 and U58, which are wired to divide the encoder signal by 25. The resulting signal, which occurs at the output of U73a then clocks another pair of counters, U64 and U65, once for each quarter-inch of travel. This pair of counters is set to count down from any binary number loaded into it from U66, a 74116 eight bit latch. This latch will contain a binary representation of the length, in units of 1/4 inch, of the present path segment. When the counters reach zero, the path section will be complete; however, the computer needs some time to calculate and load in the next two path sections so computation must start slightly before the end of the present path section. This is accomplished by comparators U62 and U63. These comparators are manually set by a DIP switch to provide a signal whenever the preset number occurs as the U64/U65 counter combination counts down. The signal alerts the computer to begin computation of the next two path segment lengths and steering angles.

The computer, having made calculations for the two subsequent path segments, will now load the distance data for each into latches U66 and U67. Data for the first path is put on the data bus and then the 'Load primary distance latch' output is selected by the computer from the Interface Board control signals. This low signal causes U66 and U67 to become transparent and upon removal of the control signal, the data is latched into both chips. Data for the second path segment is now put on the data bus and the 'Load secondary distance latch' board input will be selected to go low. At this time, only U67 will take in the data and will latch it in upon removal of this control pulse. Loading of the steering angle data into U69 and U70 will now take place in an analogous manner.

If the distance number loaded into U66 is shorter than the preset comparison number, no 'begin computation' signal will occur as U64 and U65 count to zero and therefore the computer will not reload the latches with new data. Thus the secondary latch data in U67 and U70 need to be shifted into the primary latches when the present path segment has been completed. Upon the zero count from U64 and U65, the output of U73b will go low causing monostable multivibrator U71a to output a short pulse of 0.2 μsec. The negative form of this pulse causes latches U66 and U69 to latch in data from the secondary latches while the positive version of the pulse allows the new distance data to be loaded into the U64/U65 counter pair. Now that the transfer is complete, the vehicle may continue to proceed without interruption.

Steering data at the output of U69 is sent to a DAC0800 D/A converter, U68. The output of the D/A converter ranges from 0 to 10V and is sent to U72, an LM741 op amp, to drive the steering motor. 0V corresponds to a 90°

left turn and 10V corresponds to a 90° right.

Vehicle speed is controlled by U109, a D/A converter, whose output is an analog signal between 0 and +5V. The vehicle speed data is latched into U110 with the 'Load speed latch' control signal from the Interface Board. Output from latch U110 will result in a constant analog output from U109 which will control the speed of the drive motor.

Chapter 8
The Ground Navigator Board

The Ground Navigator Board is a dedicated processor board for the ground navigation system. It is designed to compute vehicle x-y position coordinates solely from gyroscope angle and wheel encoder data.

The advantage to using the ground navigation and this processor over the optical navigation system is the high frequency in which the vehicle position may be updated. This means that current positional data is always available to the on-board computer for use in the drive software routine so that path tracking errors may be calculated and compensated for. In addition, the updated vehicle position at each beacon sighting is also stored for use by the optical navigation system's triangulation software since vehicle movement between beacon sightings must also be compensated for.

Of course, the disadvantage of the ground navigation system lies in its inaccuracy. The effect of the earth's rotation causes the gyroscope to drift over time (approximately 1/4 degree per minute). In addition, tire wear or gear slippage will affect the accuracy of the wheel encoder in the determination of distance traveled. Resolution of the encoder and gyroscope also affect the accuracy of the system. With the encoder issuing pulses for every 0.01 inches of front wheel travel, distance covered can only be determined at these intervals. (This does not mean, however, that vehicle travel is measured in units of 0.01 inches since the vehicle point is taken as being the center point of the rear axle. The actual distance covered by the vehicle point is always less than 0.01 inches for each encoder pulse unless the vehicle is traveling straight, in which case vehicle travel equals front wheel travel.) Similarly, the gyroscope has a

resolution of one quarter of a degree and therefore vehicle bearing can only be determined in angular units of this size.

8.1 The Ground Navigator Algorithm

The basic function of the Ground Navigator Board is to compute the x-y vector components of position for successive short increments of front wheel travel, Δl. For each Δl, the incremental component changes, Δx and Δy, are added to the previous x-y position coordinates to provide the current vehicle position.

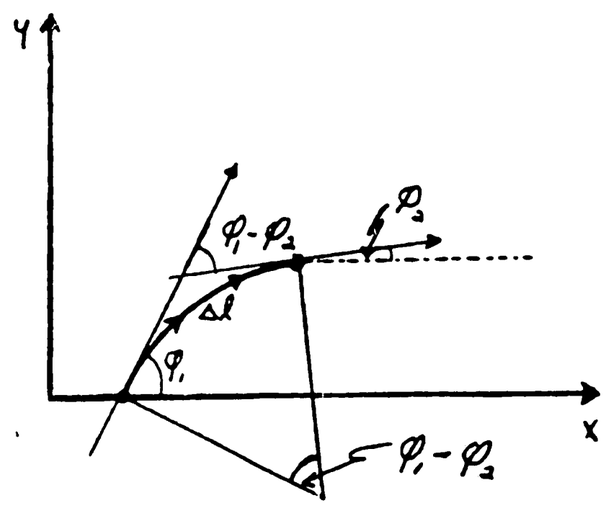


Figure 8-1: Vehicle Travel Geometry

Figure 8-1 shows that as the vehicle travels some incremental distance Δl, the vehicle bearing will change from ϕ_1 to ϕ_2 so that the Δx and Δy components describing the travel are given by

$$\Delta x=\sqrt{\Delta l^2-L^2(|\phi_2-\phi_1|)^2}\ \cos\frac{\phi_1+\phi_2}{2} \tag{8.1}$$

$$\Delta y=\sqrt{\Delta l^2-L^2(|\phi_2-\phi_1|)^2}\ \sin\frac{\phi_1+\phi_2}{2} \tag{8.2}$$

where L is the distance from the vehicle driving wheel to the vehicle

point. The square root term in the above equations translates the movement of the front wheel to the movement of the vehicle point so that the results specify the new position of the vehicle point.

A problem, however, arises from the quarter degree resolution of the gyroscope. The path increment, Δl, for the vehicle is 0.01 inches which is too short for a change in bearing to be resolved by the gyroscope. At least 14 Δl increments are needed before the gyroscope will reflect a change of $|\phi_2-\phi_1|=0.25°$. Therefore, there is no way of determining the radius of curvature of the vehicle path until a quarter degree angle change, $\Delta\phi$, occurs. In terms of $\Delta\phi=0.25°$, for a path distance $\Delta d=n\Delta l$, where n is the number of Δl increments traveled in $\Delta\phi$, equations 8-1 and 8-2 become

$$\Delta x=\sqrt{(n\Delta l)^2-L^2(\Delta\phi)^2}\cos\frac{\phi_1+\phi_2}{2} \tag{8.3}$$

$$\Delta y=\sqrt{(n\Delta l)^2-L^2(\Delta\phi)^2}\sin\frac{\phi_1+\phi_2}{2} \tag{8.4}$$

for each change of $\Delta\phi$ degrees.

Of course, the ground navigator must calculate the x and y coordinates for every Δl; therefore, an estimated path calculation must be made between $\Delta\phi$ changes. When the $\Delta\phi$ change occurs, the vehicle path radius may then be determined for the past quarter degree and thus a correction to the estimated path calculations may be made.

To estimate the vehicle path between $\Delta\phi$ changes where the path radius is unknown, a linear approximation is used. That is , the vehicle is assumed to travel in a straight line at an angle ϕ_1 as shown in Figure 8-2. The vehicle position coordinates can then be computed as

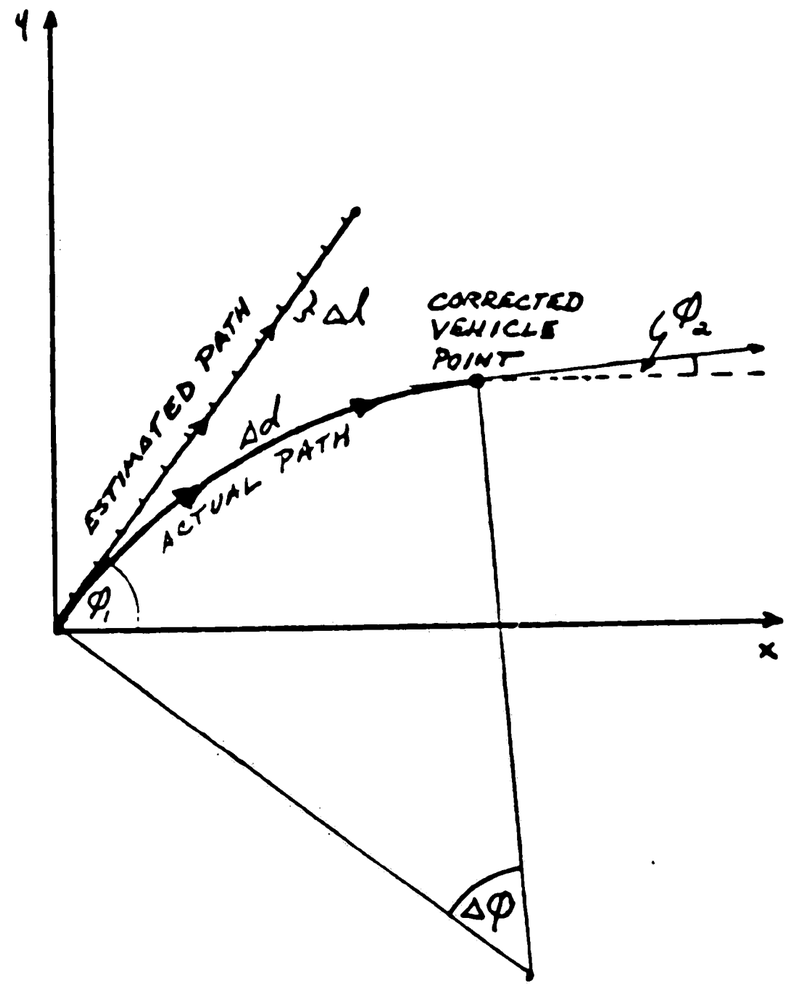


Figure 8-2: Path Estimation Geometry for $\Delta\phi$ Intervals

$$x = x_p + p\Delta l \sqrt{1 - L^2\left(\frac{\Delta\phi}{\Delta d}\right)^2} \cos \phi_1 \tag{8.5}$$

$$y = y_p + p\Delta l \sqrt{1 - L^2\left(\frac{\Delta\phi}{\Delta d}\right)^2} \sin \phi_1 \tag{8.6}$$

where x_p, y_p are the previous x and y position coordinates computed at the last $\Delta\phi$ change, p is an integer specifying the number of Δl increments traveled since the last $\Delta\phi$ change, Δd is the path distance covered in the previous $\Delta\phi$ interval, Δl=0.01 in., $\Delta\phi$=.00436 rad., L=30 in., and ϕ_1 is the present angle read by the gyroscope. Although this approximation may appear to be rather innaccurate for a small path radius, the distance traveled between $\Delta\phi$ is short

enough so that only a small error will build up. For a large path radius, the estimated path will be a close approximation of the actual path.

At the time when the $\Delta\phi$ change does occur, the vehicle will have covered a distance $\Delta d = n\Delta l$ and therefore the actual path radius traveled in the past $\Delta\phi$ may be determined so that a correction to the position coordinates can be made. The correction computation is made using

$$x = x_p + n\Delta l \sqrt{1 - L^2\left(\frac{\Delta\phi}{n\Delta l}\right)^2} \cos\frac{\phi_1 + \phi_2}{2} \tag{8.7}$$

$$y = y_p + n\Delta l \sqrt{1 - L^2\left(\frac{\Delta\phi}{n\Delta l}\right)^2} \sin\frac{\phi_1 + \phi_2}{2} \tag{8.8}$$

where n is the integer number of Δl increments covered in the previous $\Delta\phi$ and ϕ_2 is the angle seen by the gyroscope ($\phi_1 \pm 0.25°$) after the angle change. All other variables are defined as in Eqs. 8-5 and 8-6.

The newly calculated x and y coordinates represent the actual vehicle position (note, however, that the n value quantizes the distance traveled during $\Delta\phi$ and will cause a small error) and are stored as such in memory.

In summary, the ground navigator algorithm will allow the determination of vehicle position for every 0.01 inches of front wheel travel by approximating the path as being straight for a particular gyroscope angle ϕ, and then correcting for the approximation at every quarter degree angle change.

8.2 Other Board Functions

The ground navigation system is inherently not as accurate as the optical navigation system. After a period of time, error will build up in the determination of the x and y position coordinates. To keep error to a minimum, the Ground Navigator Board allows the on-board computer to send accurate position coordinates to replace the current, more inaccurate ones. The

new coordinates are obtained from the optical navigation system and will replace the current coordinates each time a position fix is obtained by the on-board computer.

The Ground Navigator Board also contains memory to store the vehicle position coordinates each time a beacon is sighted. Knowing the vehicle position at each beacon sighting is necessary for the triangulation software routine employed by the optical navigation system. This routine uses the distance traveled between beacon sightings to more accurately determine vehicle position.

The vehicle position coordinates are, of course, always available on request by the on-board computer. The current coordinates will frequently need to be accessed by the computer while performing its drive routine which compensates for vehicle tracking error away from the intended path.

8.3 Ground Navigator Board Circuitry

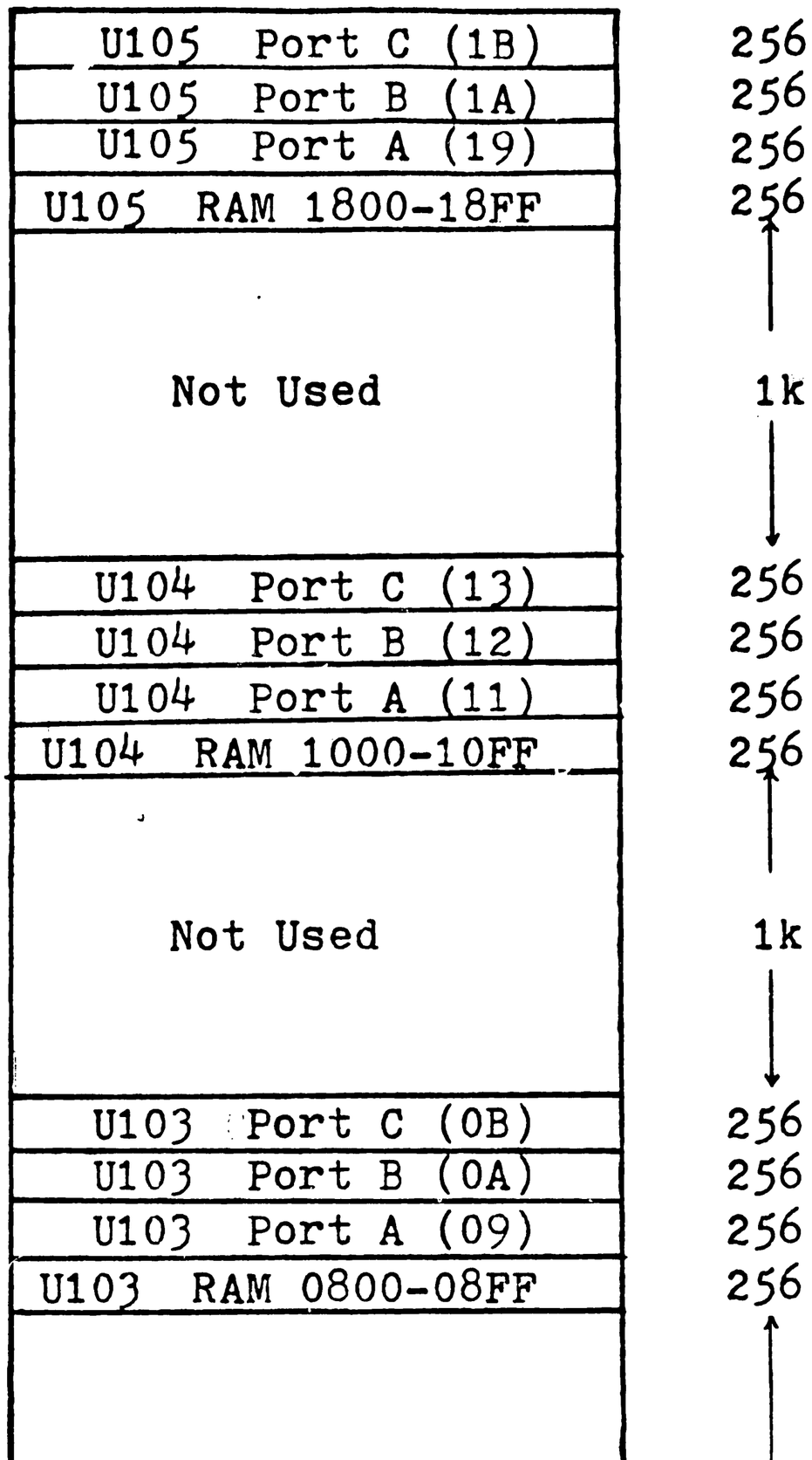
The Ground Navigator Board is a dedicated processor for the ground navigation system. It will be assumed in the following discussion that the reader is familiar with the 8085A microprocessor, around which this board is based.

First, however, some commentary on the circuits diagram. The schematics for the Ground Navigator Board may be found in Figs. A-7 and A-8. Input/output signals shown in each figure are designated by letters. Circled letters indicate references to the other figure while uncircled letters represent connections to the board edge connectors (which are irrelevant to the circuit discussion). Also, a bold line indicates the presence of a data bus whose width is marked along the line.

8.3.1 Overall Board Design

The 8085A microprocessor, U106 in Fig. A-8, is run with a 6.0 MHz crystal allowing a 3 MHz clock rate. Only thirteen address lines are needed for this system with the lower eight being connected to U107, an 8212 input/output port, to latch the low address byte, at the ALE pulse, from the microprocessor's multiplexed address/data lines. The low address byte from U107 along with A_8, A_9, and A_{10} from the microprocessor are applied to the address inputs of a 2716 EPROM, U108, to enable access to a total of 2k of ROM. The final two address lines from the 8085A, A_{11} and A_{12} are connected to U102b, a 74LS139 2-to-4 line demultiplexer. These address bits will be used for memory and I/O port selection. RAM and nine I/O ports are provided by three 8155 SRAMs with three I/O ports each; U103, U104, and U105. Each device contains 256 bytes of RAM, two 8-bit programmable ports, and one 6-bit programmable port. The memory address map and port functions for the boards are shown in Table 8-1. All low byte memory address lines and data lines are connected to the 8085A data bus.

The two low bit lines of port PC on U103 are applied to demultiplexer U102a. They will be used for selection of data from either the Gyroscope Board or the 8212 I/O ports shown in Fig. A-7. Referring to this figure, a set of six 8212 devices, U96 to U101, are shown. These chips act as a buffer between the 8085A and the on-board computer. Updated position information is supplied by the on-board in the form of two 8-bit words per x,y coordinate and loaded sequentially into U96 to U99 using 'strobe' control signals (at pin 11) generated on the Interface Board by the computer (U100 and U101 are not currently in use). The high byte of y is loaded first in U99, followed by the



PORT	FUNCTION
(09)	Input
(0A)	Input
(0B)	Output
(11)	Output
(12)	Output
(13)	Input/Output
(19)	Output
(1A)	Output
(1B)	Output

Table 8-1: Memory Address Map and Port Function Table

low byte of y in U98, high byte of x in U97, and finally, the low byte of x in U96 which generates a low signal on pin 23 indicating that the buffers are now loaded. This signal represents the 'Buffers Full' interrupt input at RST 6.5 on the 8085A. Information contained in the 8212 devices is enabled in pairs, after selection by U102a, onto a 16-bit bus connected to U103.

The 8155 device pair, U104 and U105, each have their own respective 16-bit data buses which lead to the set of 74LS244 buffers U82, U84, U86, and U88. Data present at the the input to these buffers may either be written in to the four HM6516 RAMs, U89 to U92, or may be available for direct reading by the on-board computer (by selectively enabling buffers U81, U83, U85, and U87) depending upon which operation is selected by the port PC output bits of U104 and U105. If PC0 goes low, data will be written into the RAMs, or if PC1 is set low, data will be made available directly to the on-board computer.

The four RAMs, U89 to U92, will contain the current vehicle position for each beacon sighting. Therefore the address, chip enable $\overline{E}$, and read enable $\overline{G}$ lines are the same ones that are used on the Goniometer Board (see Chapter 4). The write enable signal, $\overline{W}$, is taken from PC0 on U104 and U105 so that the position data is known to be stable before writing.

8.3.2 Interrupt Procedures

The 8085 will function on an interrupt driven basis. That is, it does nothing unless an interrupt occurs. The four interrupts and their procedure descriptions are shown in Table 8-2.

All interrupts will be serviced on a first come, first served basis with the exception of the 'Coordinate Request' interrupt which uses the TRAP input on the 8085 and is unmaskable.

INPUT	INTERRUPT	PROCEDURE
RST5.5	Beacon Sighting	Load current x,y position into RAMs U89, U90, U91, and U92.
RST6.5	Buffers Full	Read updated x,y coordinates from buffers U96, U97, U98, and U99.
RST7.5	End of Δ1	Compute and store new x,y position coordinates.
TRAP	Coordinate Request	Output current x,y position for direct reading by on-board computer.

Table 8-2: Ground Navigator Interrupts

Whenever a beacon is sighted or index pulse occurs, the 'positive half-burst or index' pulse from the Signal Conditioning Board is used to create the 'Beacon Sighted' interrupt at RST 5.5 on the 8085. This signals the Ground Navigator Board to load the present vehicle position into the HM6516 RAMs for future use during the read cycle of the optical navigation system. The procedure goes as follows:

Upon receiving the interrupt signal, the 8085 uses its two most significant address lines to select RAM U104. The current 16-bit x coordinate stored in this RAM is then output on ports PA and PB (low and high bytes, respectively). Accomplishing this, the 8085 will then set PC0, of port PC on U104, to a low state which enables buffers U82 and U84 while also lowering the write enable input, $\overline{W}$, on RAMs U89 and U90 so that the data is written to the address latched in the RAMs at this time. The y coordinate will now be loaded by selecting U105 which contains the current y position. A similar sequence of events allows PC0 of U105 to enable buffers U86 and U88 while also enabling the $\overline{W}$ input to RAMs U91 and U92. Thus the 16-bit y

coordinate will now be written at the current memory address also.

The 'Coordinate Request' interrupt pulse input to the TRAP pin of the 8085 signifies that the on-board computer requires the current position coordinates. The routine to provide these coordinates is exactly the same as the 'Beacon Sighted' interrupt routine except, in this case, the PC1 port outputs from U104 and U105 are lowered rather than the PC0 lines. This results in buffers U82, U84, U86, and U88 being enabled; however, the HM6516 RAMs will *not* be enabled. The only other difference between the two interrupt routines is that the coordinate data must remain present for a longer period of time to allow the on-board computer to select and read the data byte by byte.

The 'Buffers Full' interrupt at the RST 6.5 input of the microprocessor indicates that the on-board computer has loaded a set of updated x and y coordinates from the optical navigation system into the 8212 buffers, U96 to U99. These coordinates will be used as replacements for the current ones since they are more accurate. Upon receipt of the interrupt signal, the 8085 will select U103, which in turn enables the U96/U97 buffer pair outputs with its PC0 and PC1 output lines. U96 and U97 contain the new low and high x coordinate bytes respectively. The data is input to ports PA and PB of U103 on the 16-bit data bus. The two bytes now stored in U103 are moved into U104 to replace the current x coordinate stored there. At this point, the PC0 and PC1 outputs from U103 are changed to selectively enable the outputs of U98 and U99 which contain low and high y coordinate data bytes. This information is loaded into U103 and then moved to U105 to replace the current y coordinate.

The 'End of Δl' interrupt routine is by far the most complicated of the

interrupt procedures because of the computations involved. For every 0.01 inches of vehicle front wheel travel, this interrupt routine will be initiated and the ground navigator algorithm discussed in Section 8.1 will be carried out. This procedure begins when the interrupt signal is detected at RST 7.5 on the 8085. First, the microprocessor will select U103 to use its PC0 and PC1 to enable the gyroscope angle onto the 16-bit bus. The angle is checked to see if it has changed since it was previously examined. If it is the same, the computer will go to the subroutine to compute the position as defined by Eqs. 8-5 and 8-6, otherwise it will enter a subroutine to calculate position using Eqs. 8-7 and 8-8. In either case, the next step will be to enable U104 to examine the status of the forward/reverse input at its PC3 line. The status of this bit determines the sign of the calculated Δx and Δy position changes.

At this point, computation begins using the gyroscope data stored in U103 and coordinate data stored in U104 and U105. Cosine and sine values are obtained from a look-up table which resides in the upper ROM of U108. Values of these functions are stored for ϕ values ranging between 0 and 90 degrees at increments of quarter-degrees. Therefore, before computation begins, the gyroscope angle must be translated into this angle range. Upon completion of the required computations, the updated position coordinates are sent to U104 and U105 to replace the old ones.

Chapter 9
Remarks and Conclusions

Some remarks at this point on system limitations may provide some insight into considerations future hardware refinements.

The optical system has some limitations concerning goniometer to beacon distance. Being able to detect beacons that are far away is unimportant for this system, however, close range detection is critical, especially in docking situations. The limit to the closeness of beacons is determined by the apparent beacon widths. The counters on the Signal Conditioning Board (U23, U24, U25, and U26 in Fig. A-2) act as timers to allow the creation of the half-burst pulses. If the beacon burst exceeds the time to reach the terminal count of the counters due to the close proximity of the beacon, the counters will reset and then immediately be enabled again. This will produce half-burst pulses smaller in width than they should be causing storage of the incorrect beacon angle and width. Avoiding this situation may best be done by either an appropriate choice of physical beacon width or by increasing the count to which the counters are incremented to allow more time for long bursts.

A second consideration for close proximity beacon detection is the beam spread of the goniometer's laser. Experiments with the optical system show that the beam spread is small even for beacons reflecting the beam several feet away. Therefore, the reflected beam from a close beacon may not be able to create a burst signal since most of the light is not reflected into the goniometer's mirror lens, but rather back on the laser and its cooling fins. One solution to this problem would be to reconfigure the goniometer by moving the laser off the mirror lens to the side and replacing it by a small mirror which

reflects the beam up to the rotating mirror. This will allow more area on the lens for the beam to enter. A second solution is to use a lower quality retroreflecting surface on the beacons which spreads the beam enough to allow detection of close beacons without causing more distant beacons to undetectable.

The optical navigation system as a whole, suffers from the disadvantage that position fixes may be made only at half-second intervals. Between fixes, the ground navigation system must be relied upon and therefore accuracy is reduced. However, the possibility exists to rotate the goniometer mirror much faster, ideally to a speed in which the ground navigation system may be eliminated altogether. This scheme would produce a simpler and more accurate system, and in addition, vehicle movement between beacon sightings would become insignificant so that it need not be compensated for. There are two major drawbacks however. One is that beacons must be detectable for almost every rotation of the goniometer so that accurate navigation is possible. In many factory environments, this may be difficult due to the multitude of objects which may possibly block sightings. Secondly there is the problem of computation time. The faster the goniometer rotates, the larger the dead angle must be to allow time to compute vehicle position, and therefore less time is available to detect beacons. However, in an industrial version of this vehicle system, the goniometer would most likely be located in one corner of the vehicle rather than above it (so as to not interfere with cargo). This creates a physical dead angle of 90° which, when coupled with a fast processor, may allow the vehicle to use only the optical system.

Another limitation to be considered concerns the number of beacon sightings possible in each goniometer revolution. Many beacons should be able

to be visible to the vehicle from anywhere in the traveling environment. However, it may occur in some cases that more than the present limit of fourteen beacons may be sighted. To avoid error, the limit should be raised by adding another address line to the Goniometer and Ground Navigator Boards, however, one should be aware that this is done at the cost of more processor time.

At the time of this writing, all of the hardware for the Cyclopion vehicle system has been designed, built, and tested for proper operation. However, much of the overall system software and computation routines are currently in the development stage. For this reason, many of the vehicle specifications and limitations cannot be precisely stated at this time. Furthermore, as the software approaches the final stages of development, several modifications to the hardware may be required to accommodate new, or unforeseen software specifications. Suffice it to say that the hardware considerations presented are complete but not necessarily final.

References

1. Eberhardt, N. and Wagh, M., *Cyclopion, An Autonomous Guided Vehicle for Factory Use*; Proceedings of SPIE; Applications of Artificial Intelligence III, Vol. 635, p.536, April 1986.

2. Singh, S., *Path Planning and Navigation for a Mobile Robot*; A Thesis Presented to the Graduate Committee of Lehigh University, June 1985.

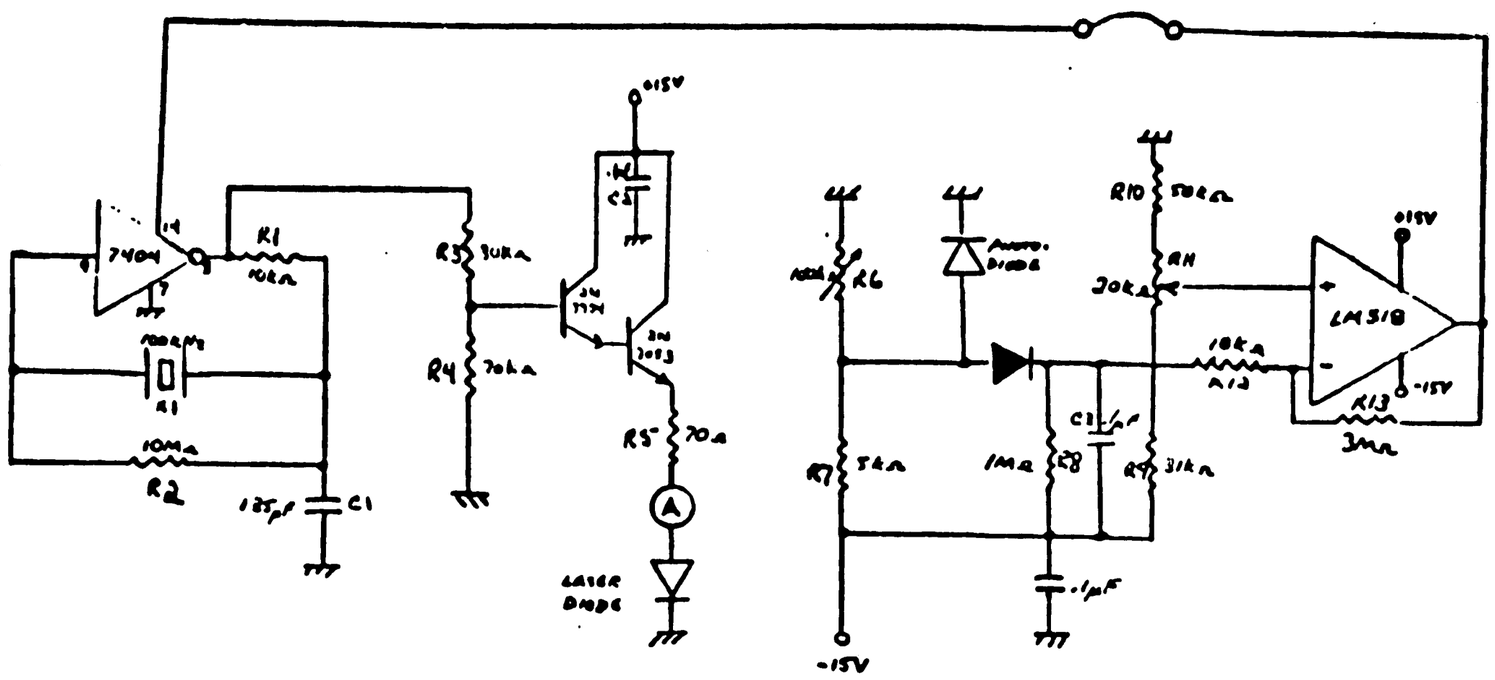
3. Marshall, C., *Documentation of Hardware Design and Theory for a Wireless AGV Navigation System*; A Documentation Submitted to SI Handling Systems, Inc., September 1985.

4. *Design and Operational Considerations for the HEDS-5000 and HEDS-6000 Incremental Shaft Encoders*; Hewlett Packard Optoelectronics Designer's Catalog, p. 9-179, 1984.

5. Julliere, M., Marce, L. and Place, H., *A Guidance System for a Mobile Robot*; 13th ISIR/Robots 7 Conference, Chicago, Ill., April 1983.

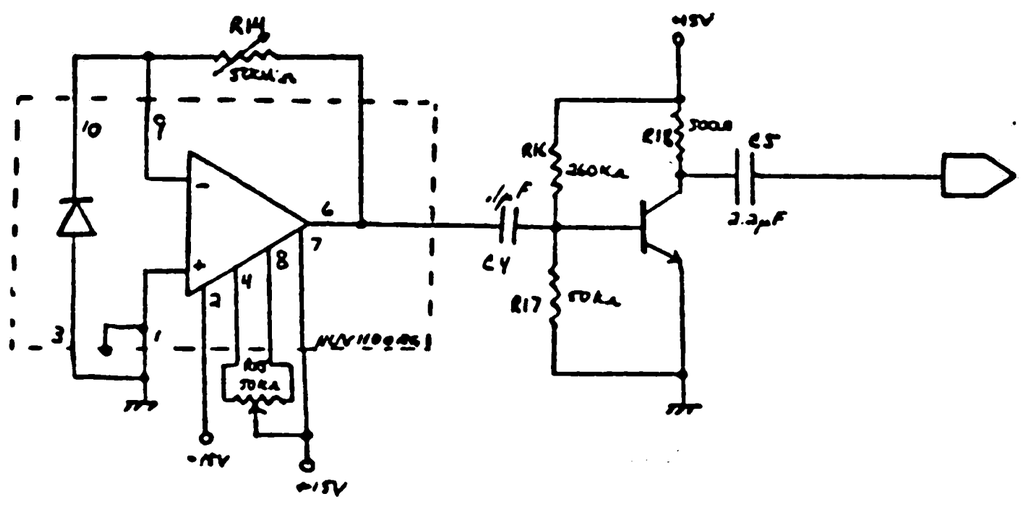
6. Jorgenson, C., Hamel, W. and Weisbin, C., *Autonomous Robot Navigation*; Byte, Vol. 11, pp. 223-235, January 1986.

Appendix A
System Circuit Diagrams



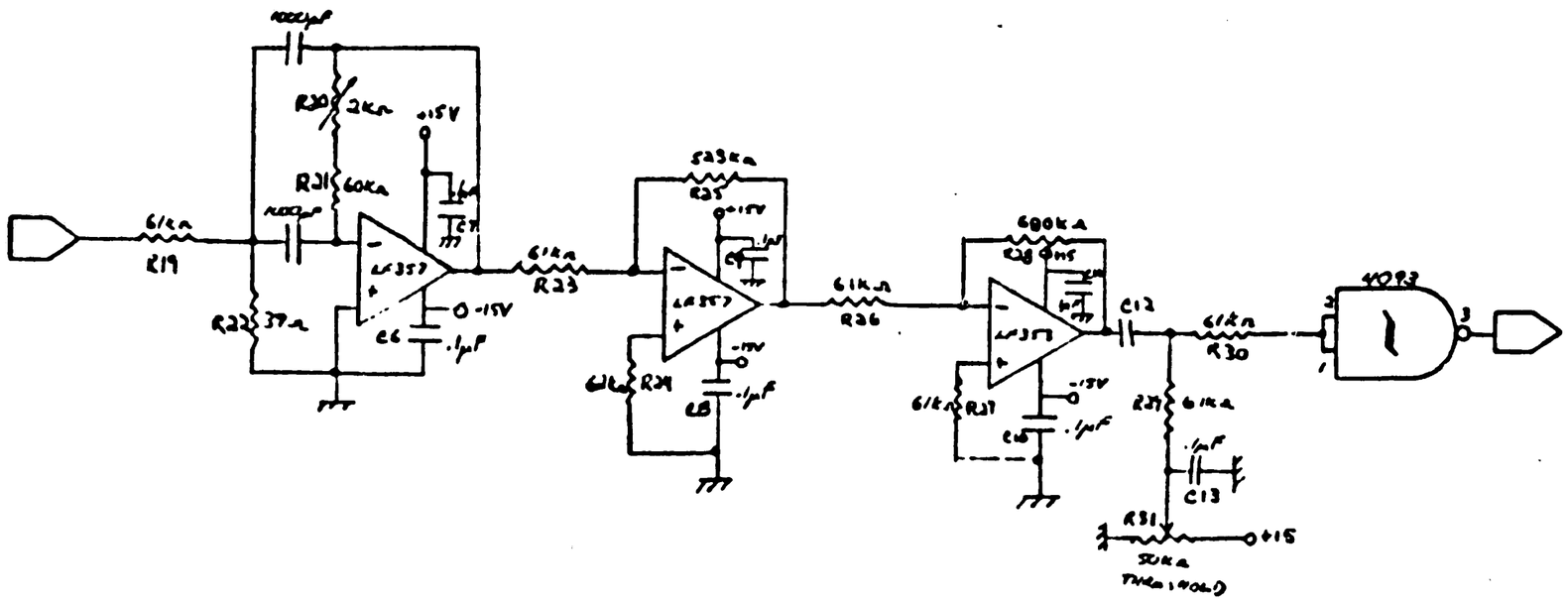
LASER DRIVER CIRCUIT

FIG. A-1(a)



PHOTODETECTOR CIRCUIT

FIG. A-1(b)



SIGNAL CONDITIONING CIRCUIT

FIG. A-1(c)

Figure A-1: Laser Driver, Photodetector, and Signal Conditioning Circuits

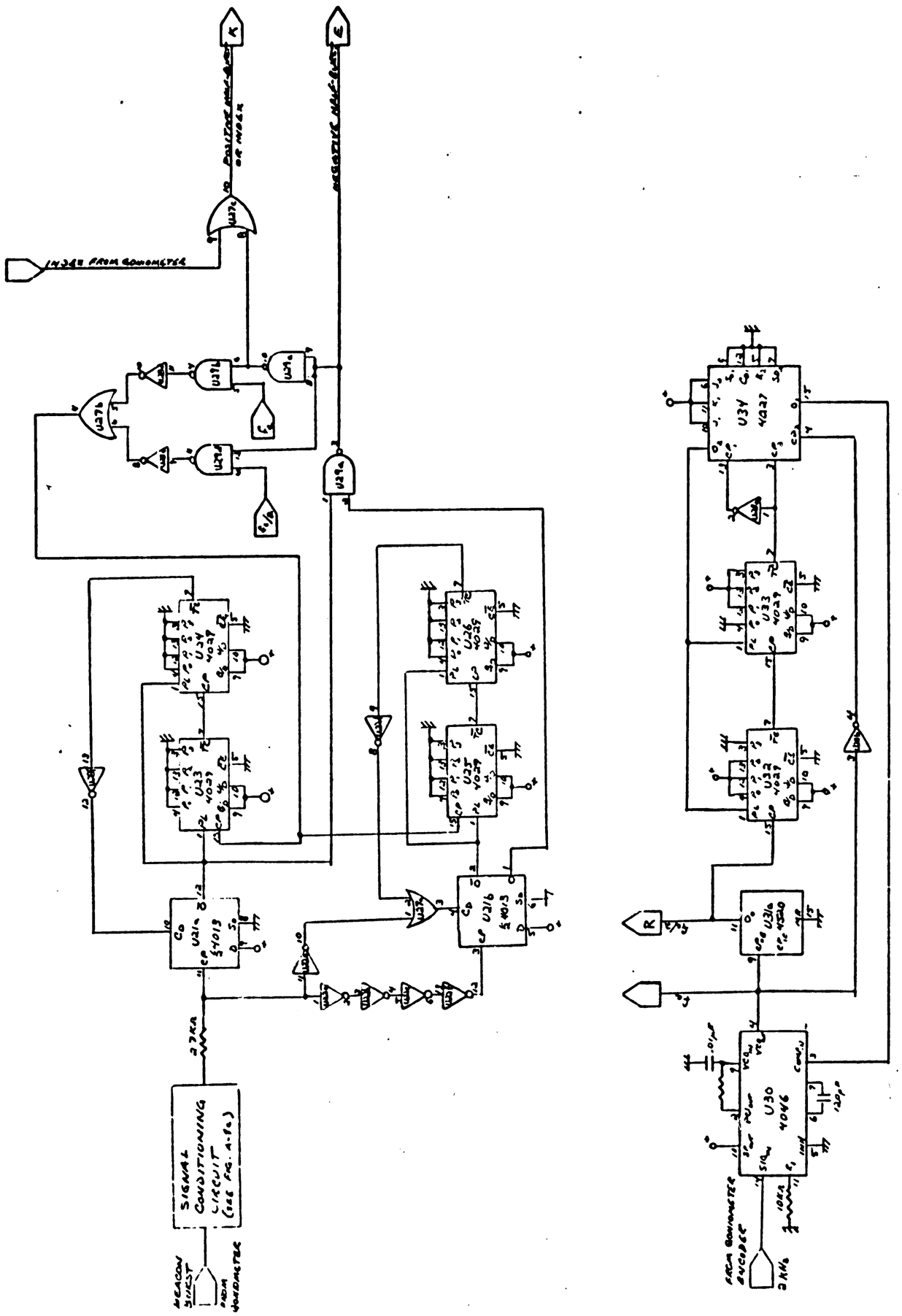


Figure A-2: The Signal Conditioning Board

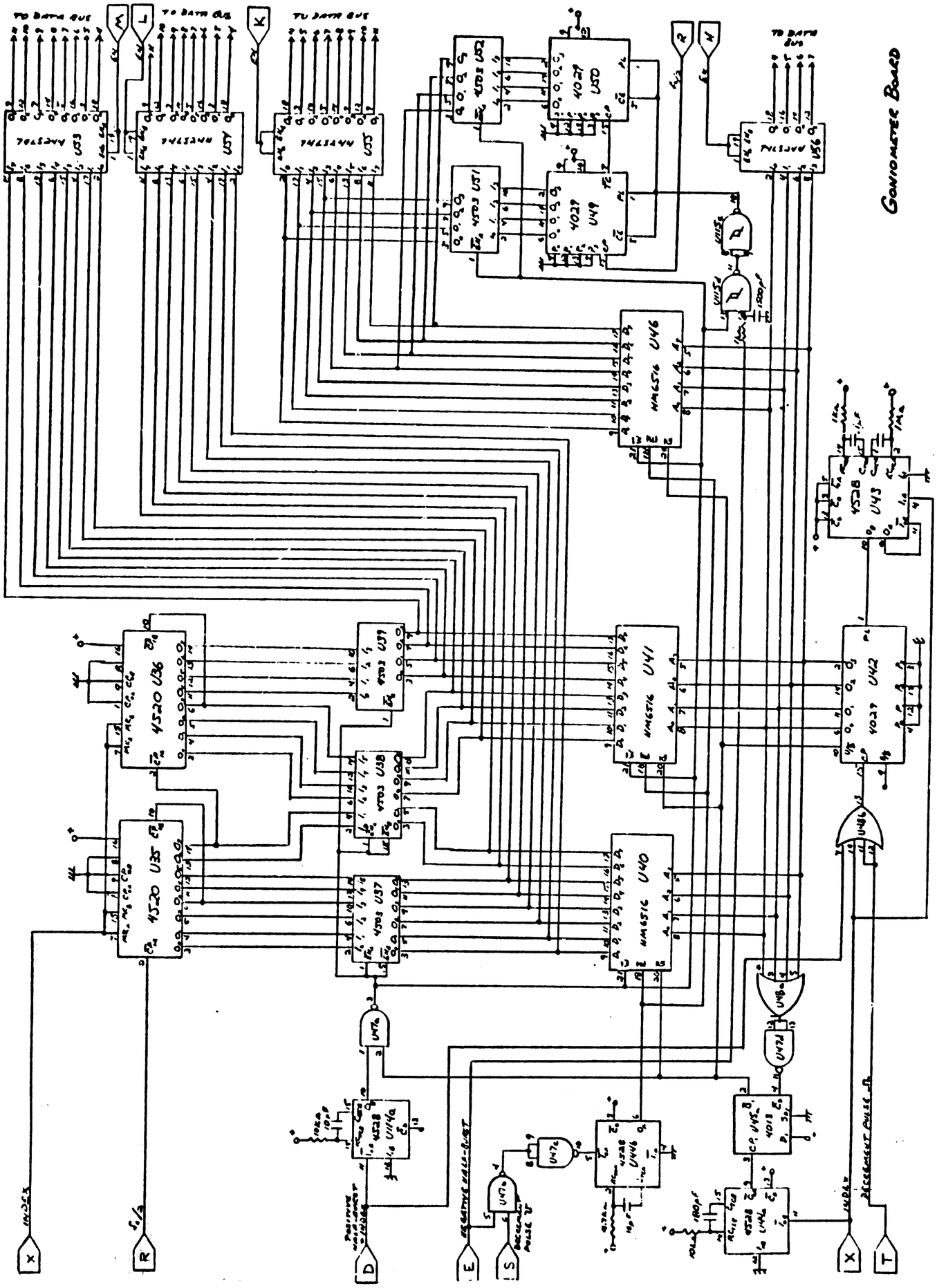


Figure A-3: The Goniometer Board

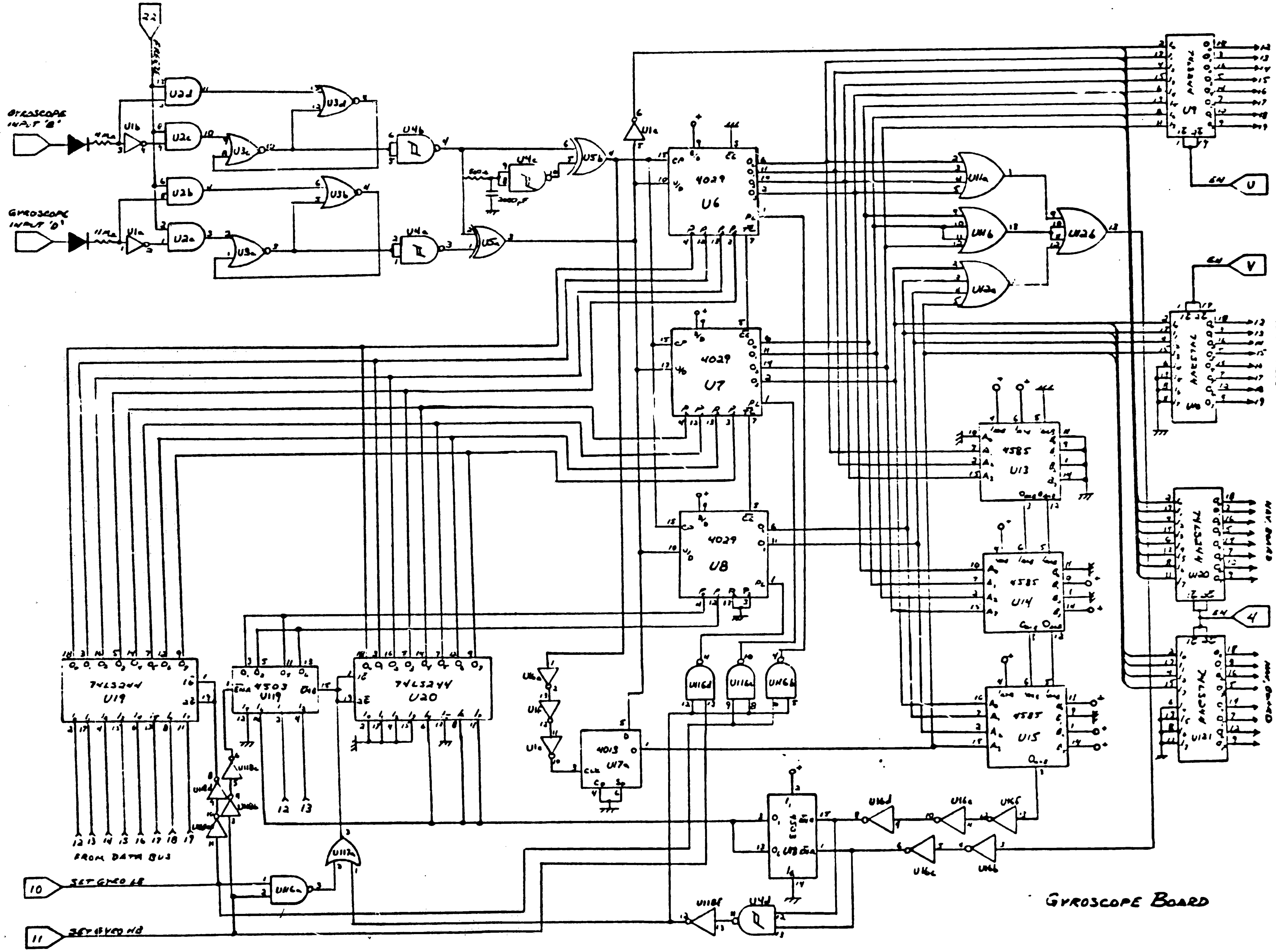


Figure A-4: The Gyroscope Board

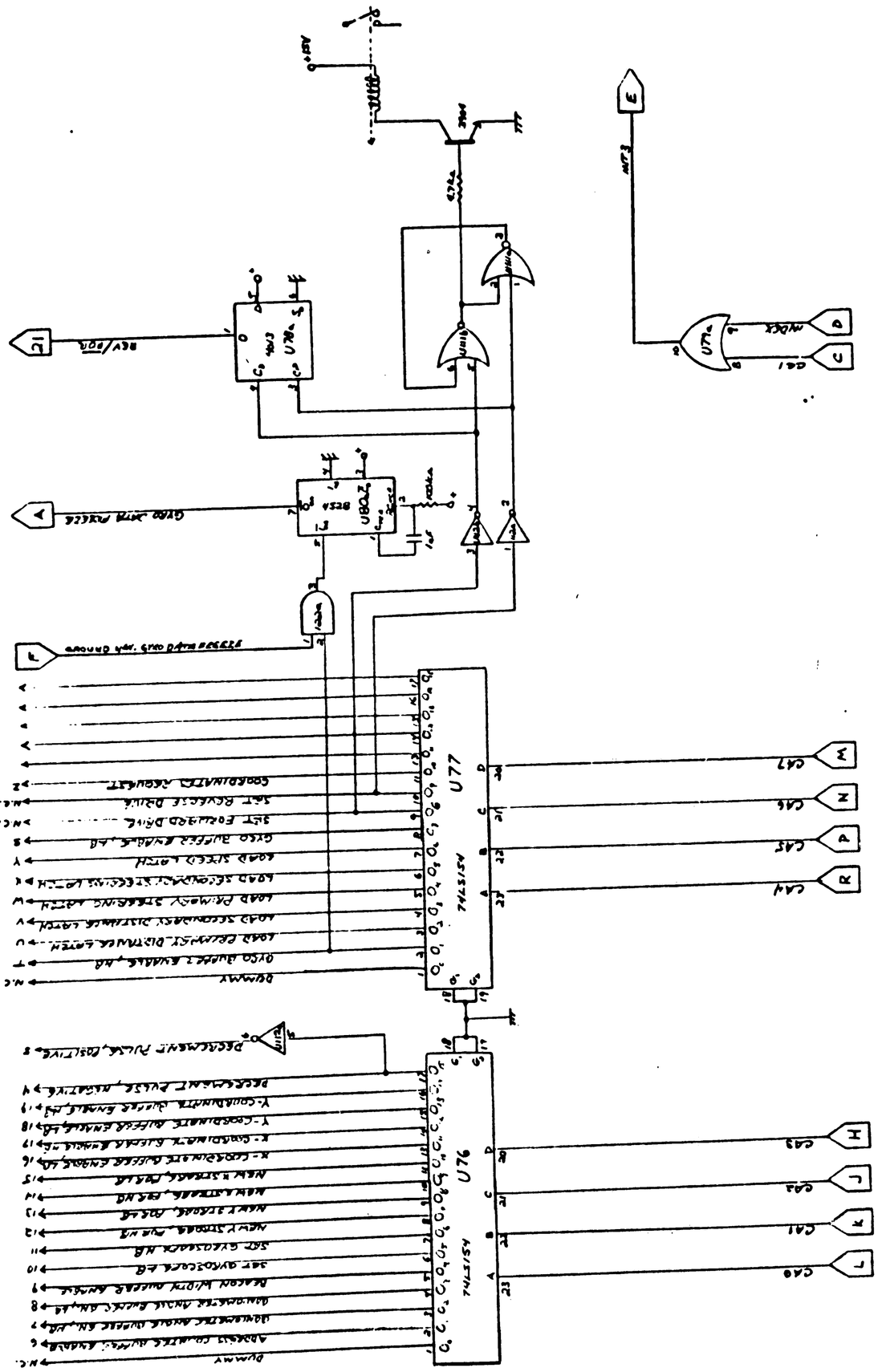


Figure A-5: The Interface Board

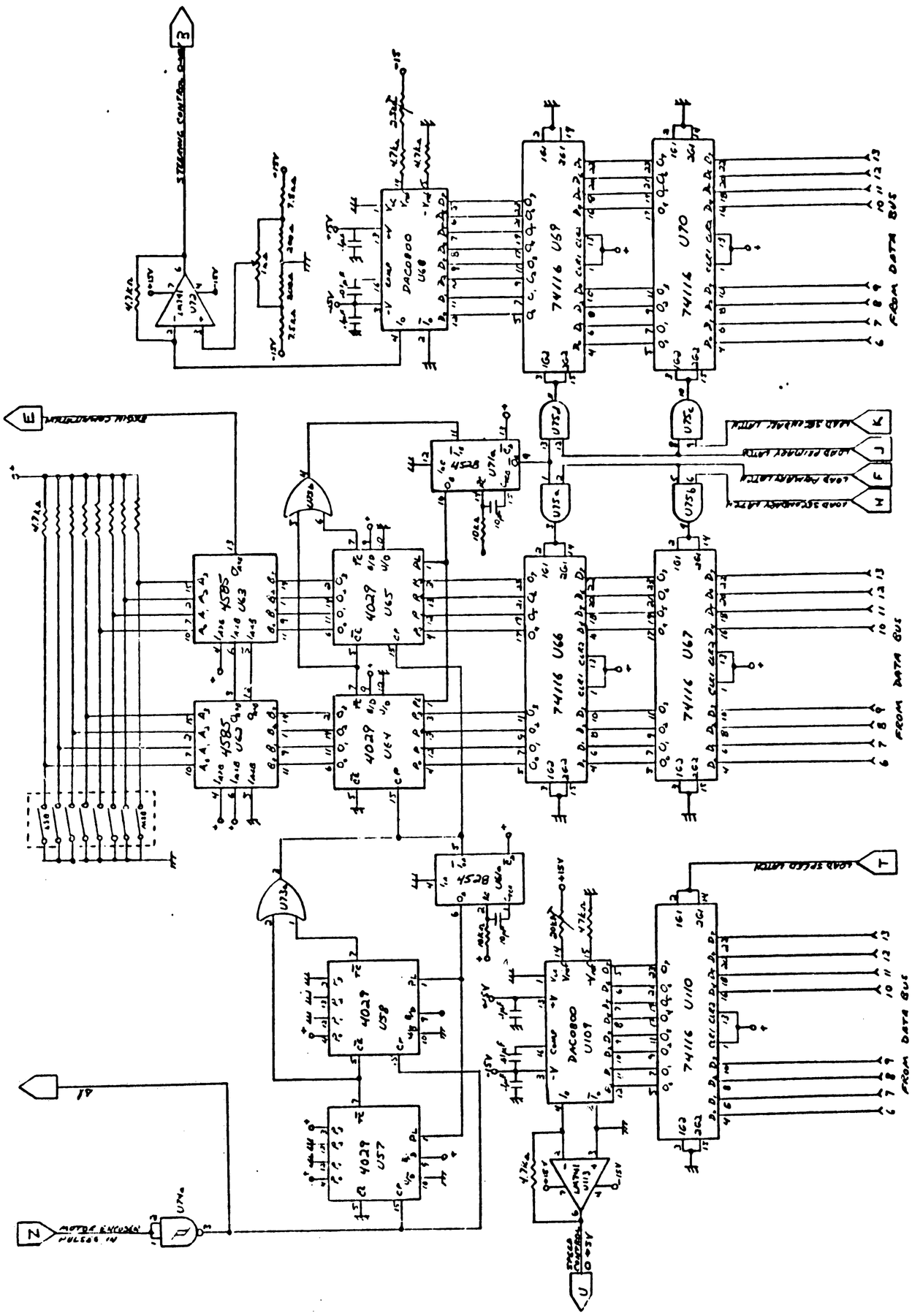


Figure A-6: The Drive Board

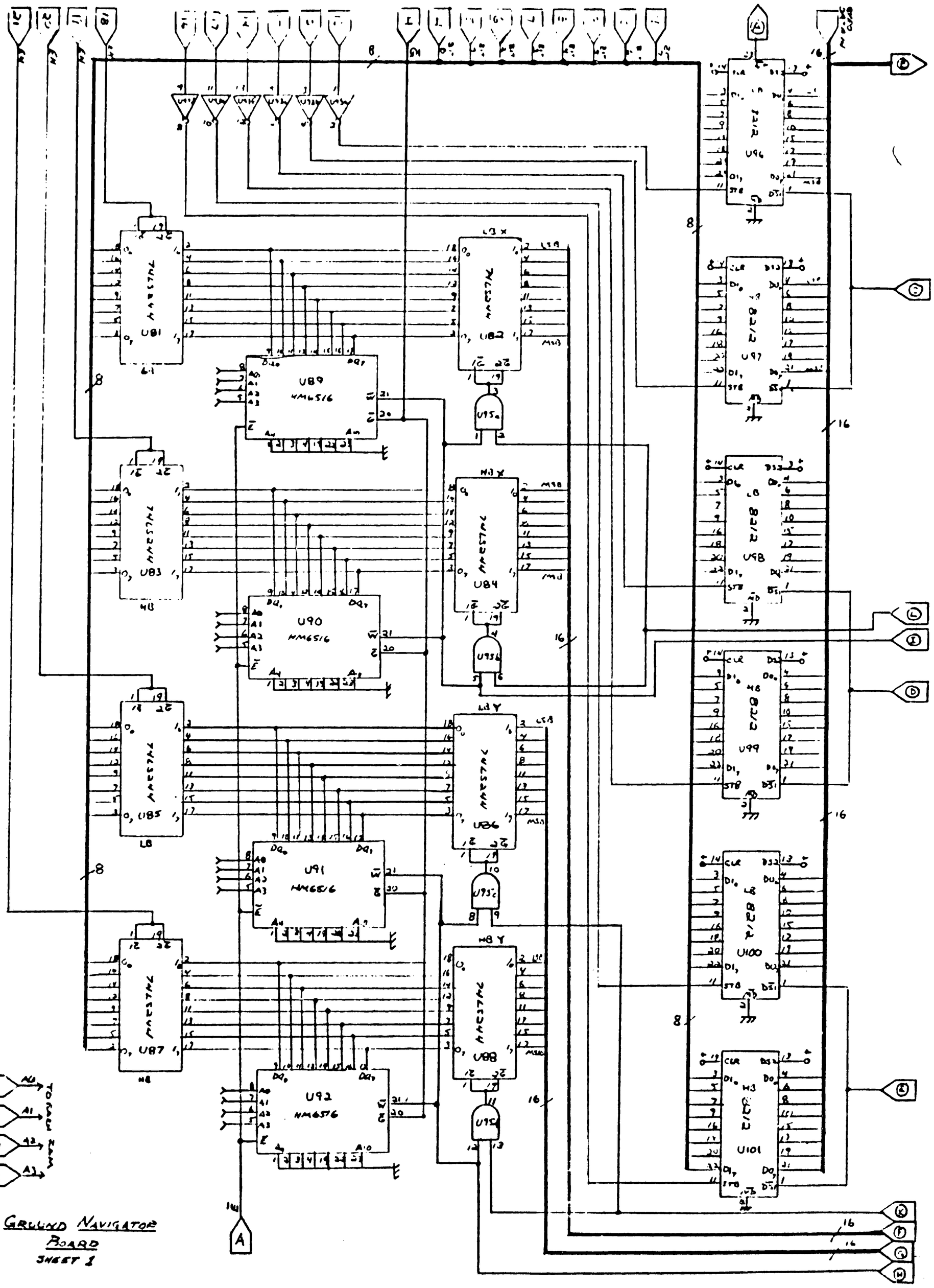


Figure A-7: The Ground Navigator Board, Sheet 1

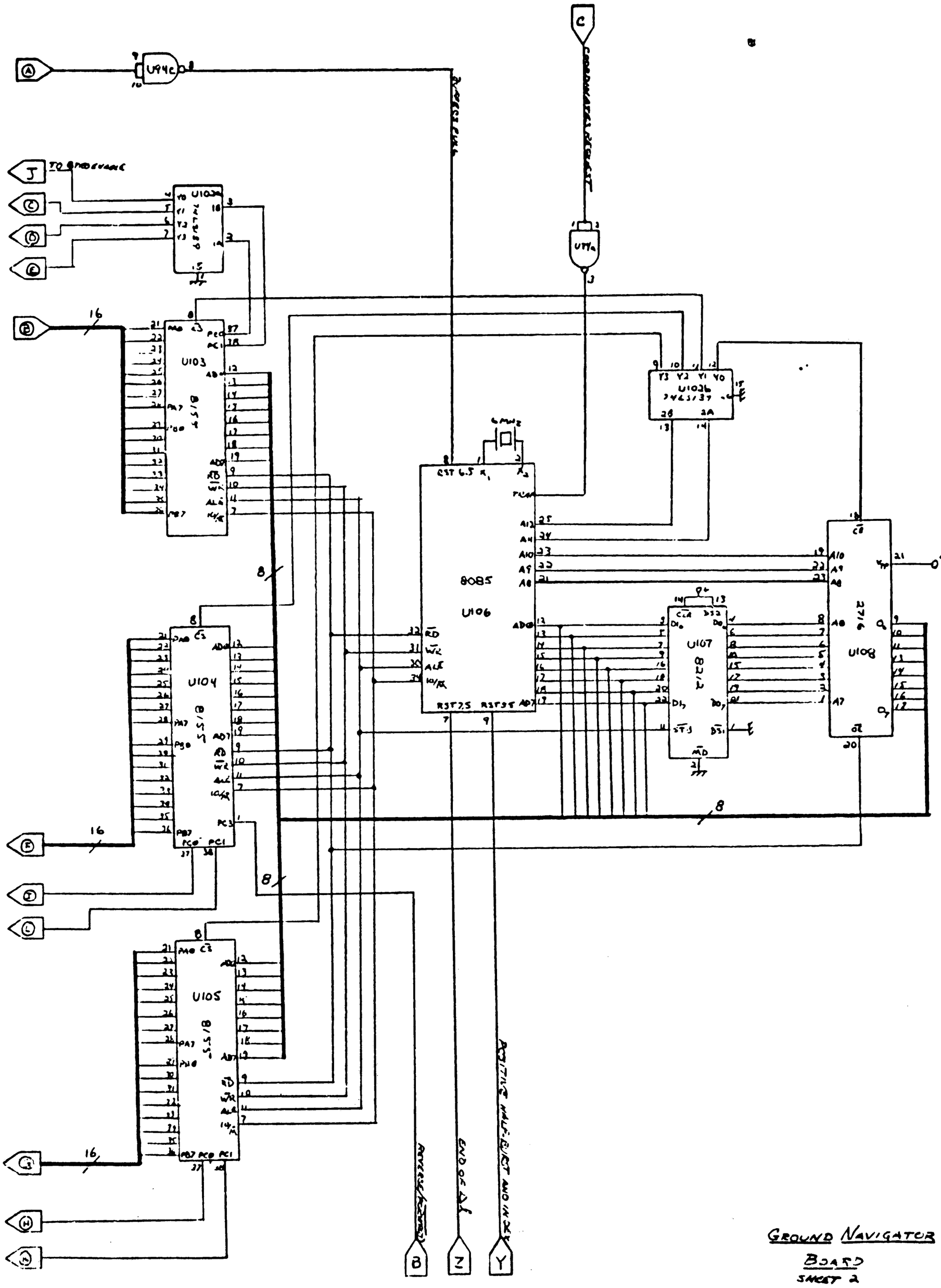


Figure A-8: The Ground Navigator Board, Sheet 2

Biography

The author, Craig Alan Marshall, was born on July 11, 1962 in Wilmington, Delaware to Mr. Albert R. Marshall and Mrs. Joan B. Marshall. His undergraduate work was done at Muhlenberg College where he received a B.S. in Physics in 1984. While at Muhlenberg, he was Vice President of the Society of Physics Students, named to the Dean's List in his junior and senior years, and also received the Eastman Kodak Award in Physics in his senior year. Graduate work was completed at Lehigh University where he was involved in research to develop an optically based, automated guided vehicle navigation system. He received an M.S. from Lehigh in Electrical Engineering in 1986.